การทำนายราคาหลักทรัพย์โดยใช้ Support Vector Machine

นายปิยทัสน์ ฉัตรวรวิทย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2559

PREDICTION OF STOCK PRICE USING SUPPORT VECTOR MACHINE

Piyatat Chatvorawit

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy in Applied Mathematics

Suranaree University of Technology

Academic Year 2016

PREDICTION OF STOCK PRICE USING SUPPORT VECTOR MACHINE

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

Thesis Examining Committee

(Asst. Prof. Dr. Eckart Schulz)
Chairperson

(Prof. Dr. Pairote Sattayatham)
Member (Thesis Advisor)

(Dr. Bhusana Premanode)
Member

(Asst. Prof. Dr. Arjuna Chaiyasena)
Member

(Asst. Prof. Dr. Benjawan Rodjanadid)
Member

(Prof. Dr. Sukit Limpijumnong)
Vice Rector for Academic Affairs and Innovation

(Prof. Dr. Santi Maensiri)
Dean of Institute of Science

ปิยทัสน์ ฉัตรวรวิทย์ : การทำนายราคาหลักทรัพย์โดยใช้ Support Vector Machine (PREDICTION OF STOCK PRICE USING SUPPORT VECTOR MACHINE) อาจารย์ที่ปรึกษา : ศาสตราจารย์ ดร.ไพโรจน์ สัตยธรรม. 304 หน้า.

ราคาหลักทรัพย์ ณ เวลาใดเวลาหนึ่งจะอ้างอิงโดยใช้ราคาปิด หรือราคาสุดท้ายที่ทำการซื้อขาย ณ ช่วงเวลานั้น ซึ่งโดยธรรมชาติของการเคลื่อนไหวของราคาปิด ข้อมูลนี้จะเป็นข้อมูลแบบ non-stationary ซึ่งไม่เหมาะที่จะนำมาใช้ในการทำนายล่วงหน้า ผลต่างของราคาปิดเป็นการแปลงข้อมูลแบบง่ายที่ทำให้ข้อมูลกลายเป็นข้อมูลแบบ stationary ได้ แต่ด้วยกฎระเบียบเรื่องราคาในการซื้อขายหลักทรัพย์ของทางตลาดหลักทรัพย์แห่งประเทศไทย อัตราการเปลี่ยนแปลงของราคาหลักทรัพย์จะเปลี่ยนแปลงไปสำหรับแต่ละระดับราคา ซึ่งอาจจะทำให้ข้อมูลผลต่างของราคาปิดอาจจะไม่เป็นข้อมูลแบบ stationary เมื่อมีการเปลี่ยนแปลงราคาจากระดับราคานึงไปยังระดับราคาอื่น โดยข้อมูลจาก SET50 จะมีหลักทรัพย์จาก 8 บริษัทที่ผลต่างของราคาไม่ผ่านการทดสอบความเป็น stationary การเคลื่อนไหวของราคาหลักทรัพย์ต้องเป็นไปตามช่วงราคาที่กำหนด ซึ่งจะแตกต่างกันตามระดับราคาของหลักทรัพย์ ตัวอย่างเช่น สำหรับระดับราคาหลักทรัพย์ระหว่าง 0.01 บาท จนถึง 2.00 บาท ช่วงราคาจะเป็น 0.01 บาท เราทำการคำนวณผลต่างของช่วงราคาของผลต่างของราคาปิดเรียกว่า “ผลต่างช่วงราคา” (tick change) ซึ่งข้อมูลที่ได้จะเป็นข้อมูลแบบ stationary และช่วงของข้อมูลจะคงที่ถึงแม้ราคาจะมีการเปลี่ยนแปลงจากระดับราคาหนึ่งเป็นระดับราคาอื่น จากข้อมูล SET50 จะมีเพียง 2 บริษัทเท่านั้นที่ไม่ผ่านการทดสอบความเป็น stationary ดังนั้นวิธีการแปลงข้อมูลที่เราเสนอขึ้นมาจะเหมาะสำหรับการแปลงข้อมูลหลักทรัพย์มากกว่าการใช้ผลต่างของราคา

นอกจากนี้ยังทดสอบอินดิเคเตอร์อย่างง่ายจากข้อมูลที่แปลงข้างต้น คือ SMA และ EMA จากผลต่างของราคาและผลต่างของช่วงราคา จากการทดลองกับข้อมูลหลักทรัพย์ของ SET50 อินดิเคเตอร์ SMA จากข้อมูลที่แปลงให้ผลที่ดีกว่าอินดิเคเตอร์ที่เป็นที่นิยมอย่าง MACD

สำหรับการทำนายราคาหลักทรัพย์ โมเดลที่เราสร้างขึ้นโดยใช้ SVM จากข้อมูลผลต่างของราคา และผลต่างของช่วงราคาจะให้ผลที่ดีกว่าการใช้ราคาปิดโดยตรง และให้ผลที่ดีกว่า neural network ยิ่งไปกว่านั้น ARIMA ที่ใช้ข้อมูลผลต่างของราคา และการใช้ราคาปิดวันนี้เป็นราคาทำนายจะให้ความคาดเคลื่อนสูง

สาขาวิชาคณิตศาสตร์

ปีการศึกษา 2559

ลายมือชื่อนักศึกษา Piyadol Chadvorant

ลายมือชื่ออาจารย์ที่ปรึกษา P Sattayatham

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

PIYATAT CHATVORAWIT : PREDICTION OF STOCK PRICE USING SUPPORT VECTOR MACHINE. THESIS ADVISOR : PROF. PAIROTE SATTAYATHAM, Ph.D. 304 PP.

STOCK PRICE PREDICTION / SUPPORT VECTOR MACHINE / SUPPORT VECTOR REGRESSION / TECHNICAL ANALYSIS / EMPIRICAL MODE DECOMPOSITION

A stock price at any particular time is represented by its closing price, i.e., the last traded price in the stock market prior to that particular time. By their nature, closing prices are non-stationary data, and thus not suitable for prediction. Considering closing price differences is a simple transformation which makes the data stationary. However, by the price movement rules of the Stock Exchange of Thailand (SET), the range of possible closing price changes varies for different price intervals; there are some rules governing limits and patterns for price movements. Thus, closing change data may not be stationary when moving from one price interval to another. From our dataset from SET50, there are eight stocks that fail the test of stationarity. Tick size is the smallest amount by which the stock price can change. For the SET market, tick size varies for various price intervals, e.g., tick size is 0.01 Baht per step for prices between 0.01 Baht and 2.00 Baht. We calculate the number of ticks different in price change, called tick change, which is also a stationary data and its range remains the same for different price intervals. With the same dataset from SET50, there are only two stock that fail the test. So our transformation to tick difference is more suitable to transform stock data from SET into stationary data.

We also present an indicator based on closing change and tick change, i.e.,

SMA and EMA from closing change and tick change. With our dataset from the SET50, our SMA from closing change and tick change give a better profit than the popular MACD indicator.

For stock price prediction, our SVM model based on closing change and tick change outperforms than the closing price model, and the neural network model. Additionally the ARIMA model with closing change and today's price as predicted price model from closing price present a high error.

School of Mathematics

Academic Year 2016

Student's Signature _____

Advisor's Signature _____

Co-advisor's Signature _____

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Professor Dr. Pairote Sattayatham, and my co-advisor, Dr. Bhusana Premanode, for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my Ph.D study.

Beside my advisor and co-advisor, I would like to thank the rest of my thesis committee: Asst. Prof. Dr. Eckart Schulz, Asst. Prof. Dr. Arjuna Chaiyasena, and Asst. Prof. Dr. Benjawan Rodjanadid, for their perceptive comments and encouragement, but also for the hard questions which incited me to widen my research from various aspects.

Special thanks to a staff member at the School of Mathematics, Ms. Anusorn Rujirapa, for her help in dealing with university administration issues.

I thank my friends, Dr. Amornrat, Dr. Suntorn, Dr. Nawarat, Dr. Tosaporn, Dr. Nop, Dr. Sasithorn, Wiparat, Dr. Tippathai and all member of our Mathematics family at SUT, without whom I could not have had wonderful time at SUT. Thanks for eating, smiling, and laughing with me, and also helping me solve any problems in my life.

Piyatat Chatvorawit

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

Page

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF SYMBOLS

$buyVol_i$	the buying volume of i^{th} day
$change_i$	the closing change of i^{th} day
$close_i$	the closing price of i^{th} day
$direction_i$	the direction of price movment of i^{th} day
ema_i	the EMA of i^{th} day
$ema10_i$	the 10 days EMA of i^{th} day
$ema60_i$	the 60 days EMA of i^{th} day
$ema250_i$	the 250 days EMA of i^{th} day
$ema10Change_i$	the 10 days EMA change of i^{th} day
$ema60Change_i$	the 60 days EMA change of i^{th} day
$ema250Change_i$	the 250 days EMA change of i^{th} day
$imf_{(i,n)}$	the value of n^{th} day of i^{th} IMF
rsi_i	the RSI value of i^{th} day
sma_i	the SMA of i^{th} day
$sellVol_i$	the selling volume of i^{th} day
$tickCount_i$	the number of tick of the closing price of i^{th} day
$tickChange_i$	the tick change of i^{th} day
vol_i	the trading volume of i^{th} day

LIST OF ACRONYMS

ADF	Augmented Dickey-Fuller
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
BPNN	Back Propagation Neural Network
EMA	Exponential Moving Average
EMD	Empirical Mode Decomposition
HHT	Hilbert-Huang Transform
IMF	Intrinsic Mode Function
KPSS	Kwiathowski-Phillips-Schmidt-Shin
MACD	Moving Average Convergence Divergence
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NN	Neural Network
R^2	R-squared
RBF	Radial Basis Function
RSI	RElative Strength Index
SD	Sum of the Difference
SET	Stock Exchange of Thailand
SMA	Simple Moving Average
SVM	Support Vector Machine

LIST OF ACRONYMS (Continued)

SVR Support Vector Regression

CHAPTER I

INTRODUCTION

A stock market is where companies' shares are traded. It is not really a place or a building anywhere; it is just a network of trading transactions. The origin of the stock market can be traced back to around the 12^{th} century in France, at that time people traded with agricultural debts in their communities. At the present time, there are stock markets in almost every country, and the United States' stock market is the world's largest market. (World Federation of Exchanges, 2012)

Shares are traded at agreed prices between two parties, so a stock's price can change rapidly and dramatically depending on the demand and supply in the stock market. A change in the share's price affects the company's value directly, and a company can raise funds easily by selling additional shares to the public. There are many people and companies making profit from buying and selling shares in the market. All of these need strategies to increase the chance to make profit and reduce risk. The stock index was introduced as a method to measure the market value.

The first stock index was introduced in 1896 by Charles H. Dow, a financial journalist (Ray, 2012). He developed the "Dow Jones Industrial Average" which was an average of the top 12 stocks in the market at that time; most of these stocks were in the industrial sector. After that, he published more works, which formed the basis of "Dow Theory" later. This theory is the fundament of technical analysis that is widely used as a tool to analyze the stock market nowadays.

Many indices have been developed to measure each segment of the stock market, i.e., the S&P index, or the Nasdaq index. These indices are used as benchmarks by investors to gauge and plan their investments in the markets.

Most of the research in forecasting stock focuses on predicting stock indices. On the other hand, predicting the stock prices of individual companies hasn't attracted much attention from the public, as there are so many different factors for each company. An individual company's share is affected easily by external factors, e.g., news, politics, rumors, or even stock prices of other companies.

The main goal of this research is to predict individual stock prices. Therefore this study proposes a prediction model for forecasting a single stock price.

1.1 Motivation

In the stock market there are basically two ways for making profit. One is by the capital gain of a stock, i.e., buying low and selling high. Another one is by the dividend yield from a stock. These two require different strategies. For the latter, one needs to grasp a reasonable knowledge of that particular company's business, and also understand the situation in the market to be able to adjust one's portfolio accordingly to decrease the risk. For the former, depending on the length of holding the stock between buying and selling, one may only need a bit of knowledge of that particular company, or none at all. The holding period may range from a few seconds to several years. Day-trading is used to refer to a holding period which only last for a few days to a week. This requires a good prediction model to reduce the risk. On the other hand, for a longer holding period, we believe that it is better to analyse/forecast the stock price based on that particular company's business, and other information related to the company, e.g., market growth, business plan, etc.

We focus on short term trading in this study. There are many methodologies in forecasting. Machine learning is one of the most recent developed techniques. It has many advantages over the other methods, the need for less human intervention may be its most beneficial, since humans can have bias in their judgement, i.e., they may create and adjust a prediction model to produce the preferred outcome.

In the area of machine learning, the support vector machine (SVM) has become quite the popular over the recent years. It was developed by Vapnik et al. (1995). Its popularity has been increasing due to many attractive features and its promising empirical performance in a variety of applications such as pattern recognition, regression estimation, time series prediction etc. It has many advantages over other machine learning techniques as we are going to discuss in detail later. Combined with technical analysis, one of the classical technique traders use to analyze stock price, we believe it will help produce a better forecasting model for an individual stock price.

1.1.1 Literature Review

There are many researches and developments attempting to predict the stock indices. These have continued for several decades now. Basically, there are two main groups of researchers in the area of stock index forecasting. The first group of researchers believes that the stock index can be foreseen, and they have developed many mechanisms to predict the market. The researchers in the second group believe that it is nearly impossible to forecast the market, and that a stock index follows a random walk, which implies that the best prediction you can have for tomorrow's value is the value of today.

In stock index prediction, E. Kalyvas (2001) generally categorizes the re-

searches into four categories as follows:

1. Technical Analysis Method (also known as chartists). It uses a historical data of the stock index to plot graphs. This graph is then used to identify the trend of the market by various formulas.

2. Fundamental Analysis Method. It attempts to find the intrinsic value of the stock by analyzing the business plan, financial information, and all related information.

3. Traditional Time Series Forecasting. It tries to create prediction models to determine the patterns of historical data. These models can be divided into two groups: the univariate and the multivariate regression models, depending on whether they use one or more variables to approximate the stock market.

4. Machine Learning Method. There are many methodologies that have been developed in this group. In general, they use a set of sample data to determine the function that has generated the data, and then this function is used for prediction. Usually, all of the learning and calculations are done by machine.

Each stock market has its unique characteristic. Two stock markets may have different factors that affect their indices. A prediction methodology that works well for one stock market may not do well in other stock markets. In order to get an optimal result from prediction, we need to study that particular stock market. In our case, we are going to work on the stock market of Thailand.

In Thailand, the stock market (Leoywanichjalearn, 2009) is managed by the Stock Exchange of Thailand (SET). As of 31 December 2011, the Stock Exchange of Thailand had 545 listed companies with a combined market capitalization of

฿8,490 billion. The indices of the stock exchange are the SET Index, the SET50 Index, the SET100 Index, the SETHD Index, the Industry Group Index, and Sectoral Indices. The SET Index is calculated from all stocks trading on the SET, while the SET50, SET100, and SETHD indices are calculated from selected stocks that pass the criteria, e.g., large market capitalization, high liquidity, the distribution of shares to minor shareholders. The Industry Group Index and Sectoral Indices are calculated from all the stocks in the same industry or sector, so there are several indices linked to this index.

There have been several researches in predicting the SET index. Neural networks are quite a popular technique for forecasting from their learning abilities. They do not require human intervention in extracting and formulating any functions or models for prediction; they can learn by themselves from the training data. However, they have some significant drawbacks. They suffer from local-optima as they may converge to a local minimum instead of the global minimum. The convergence rate is quite slow (Chaigusin et al., 2008; Sutheebanjard and Premchaiswadi, 2010). An evolution strategy is also popular in forecasting the SET index. It is an optimization technique based on a natural evolution process (Rimcharoen et al., 2005; Sutheebanjard and Premchaiswadi, 2010). There is also a research on using soft computing techniques to forecast the SET index. While traditional computing that we are familiar with, called hard computing, deals with precision, certainty, and exact solution; soft computing on the other hand has been invented to work on imprecision, uncertainty, and approximation. Generally, the three main areas in soft computing are fuzzy logic, neural networks, and probabilistic reasoning (Chaigusin et al., 2008).

Recently, there is increasing interest in applying SVM for financial forecasting. F.E.H Tay and L. Cao (Tay and Cao, 2001) compared the performance

of SVM and a multi-layer back-propagation neural network (BPNN) on forecasting five real futures contracts from the Chicago Mercantile Market. The results showed that SVM does better than BPNN. W.H Chen and J.Y. Shih (Chen and Shih, 2006) did an experiment to compare the performance of SVM and BPNN for forecasting the six major Asian stock markets, e.g., the Nikkei 225, the All Ordinaries, the Hang Seng, the Straits Times, the Taiwan Weighted, and the KOSPI. They found that the results were varied by the markets, e.g., SVM did better for some markets, while BPNN was better in other markets. M. Kumar and M. Thenmozhi (Kumar and Thenmozhi, 2009) did a study on comparing SVM and other traditional methods for forecasting the direction of the S&P CNX NIFTY Market Index. Their experimental results showed that SVM outperformed random forest, neural network, and other traditional methods. W. Huang, Y. Nakamori, and S.Y. Wang (Huang, Nakamori, and Wang, 2005) studied the ability to forecast the financial movement direction of the NIKKEI 225 index. Their experimental results showed that SVM outperformed the other classification methods. Y. Kara, M.A. Boyacioglu, and O.K. Baykan (Kara, Boycioglu, and Baykan, 2011) did a research on applying artificial neural network (ANN) and SVM in predicting stock index movement for the Istanbul Stock Exchange. Their results showed that an ANN model did better than the model of SVM.

1.2 Objectives

The objectives of the research are the following:

(1) To investigate the existing methodologies for stock forecasting.

(2) To derive an improved (or new) methodology by employing SVM, Empirical Mode Decomposition (EMD), and technical analysis approaches.

(3) To optimize the approach in (2) for stock price prediction of the companies in SET.

(4) To compare our methodology with the existing methodologies.

1.3 Organization

This thesis is organized into six chapters as follows. Chapter I, the motivation behind this research has already been described, as well as its objectives and organization. Chapter II describes the theoretical background related to stock prediction and SVM. Chapter III discusses the methodologies used in this study. Chapter IV shows and discusses the stock data that are used in this research. Chapter V describes the proposed model, along with the experimental results. Chapter VI provides a conclusion, and possible improvement to the model.

Additionally, in the Appendix, programme files, experimental results, and all Figures and Tables that not shown in the previous chapters are presented.

CHAPTER II

PRELIMINARIES AND LITERATURE REVIEW

In this chapter, the basic definitions and concepts related to technical analysis, support vector machine (SVM), and empirical mode decomposition (EMD) are presented.

In stock prediction, technical analysis has been quite a popular technique for several decades. It is a graph technique developed based on several market assumptions derived from the concepts of Dow theory. Many graph types, overlays, and indicators have been developed to help traders identify market trends, and buying-selling point.

A support vector machine (SVM) is one of the popular technique in machine learning. It has been developed for use in classification and regression tasks. Its application in the area of financial forecasting has gained attention recently. Its main concept is similar to the other machine learning techniques by training the model with the dataset, then use the trained model for forecasting.

An empirical mode decomposition (EMD) was developed as a part of the Hilbert-Huang transform (HHT). It is a decomposition method used to transform any given data into Intrinsic Mode Functions (IMFs).

2.1 Preliminary Concepts

In this section, we are going to explain the basic terminologies, techniques, and methodologies relate to this study.

Time Series

A time series is a series of data collected in timely order. Usually, it is recorded at regular time intervals, e.g., every second, minute, ... Thus it is a sequence of discrete-time data. The common example of the time series is most of data related to economic or business, e.g. stock price, stock index, exchange rate.

Stationary Time Series

A stationary time series is a series of data whose statistical properties are constant over time, i.e., mean, variance, autocorreclation, etc. are the same at any time. Many prediction models are based on the assumption of stationary data. Usually, the time series can be transformed into stationary data by mathematical transformations.

2.2 Dow Theory

The theory was derived from 255 Wall Street Journal editorials written by Charles H. Dow (1851–1902). Dow theory can be summarized into six basic tenets (Ray, 2012):

(1) The market has three movements.

(a) Main movement. It is the major trend of the market. It can last for several years.

(b) Medium swing. It can last for several months. The market usually changes from 33% to 66% of the primary price change since the previous medium swing or start of the main movement.

(c) Short swing. It can last from hours to a month.

(2) Market trends have three phases.

(a) Accumulation phase. In this phase, investors who know the inside information are actively buying (or selling) a stock against the general opinion of the market. During this period, the stock price does not change much because these investors are in the minority demanding stock that the market at large is supplying.

(b) Absorption phase (or public participation). In this phase, a rapid price change occurs as the market catches on to those investors (in (a)). It usually happens when trend followers and other technical oriented investors participate.

(c) Distribution phase. The investors (in (a)) begin to distribute (or collect) their shares to the market in this phase.

(3) The stock market discounts all news. When there is new information available to the public, stock prices will change to reflect this information. This is similar to one of the concepts of the **efficient market hypothesis**.

(4) Stock market averages must confirm each other. In the period of Dow, the industrial and transportation (railway) sectors were the most important sectors in the US market. The averages of these two sectors should be moving in the same direction as economies of both sections rely heavily on each other. If these two averages are moving in different directions, it is a sign that a change is going to happen soon.

(5) Trends are confirmed by volume. When prices move on low volume, there are many reasons to explain these. On the other hand, if price movements

are accompanied by high volume, Dow believed this represented the "true" market view.

(6) Trends exist until definitive signals prove that they have ended. Dow believed that trends existed despite "market noise". Markets might temporarily move in the direction opposite to the trend, but they will soon resume the prior move. But this can be a signal of a new trend. Technical analysis tools are usually used to identify this, but different investors can interpret them differently.

2.3 Technical Analysis

Technical analysis is a forecasting technique of price movements based on the historical data of price. Generally, it uses charts and indicators to identify the behavior of the stock and predict a price trend. It is based on three assumptions:

(1) The market discounts everything. This is the same concept as in Dow's theory. Stock price usually changes to reflect all relevant information.

(2) Prices move in trends. After a trend is formed, the future price movement is likely to follow this trend. Most of technical trading strategies are based on this assumption.

(3) History tends to repeat itself. Generally, market participants tend to react similarly to similar situations of a market over time.

There are many types of charts that are used in technical analysis. Basically, it is a price chart with a time frame. A time unit can be minute, hour, day, week, month, quarter or year depending on the purpose of chart. Usually, a short time frame is used for forecasting short-term price movements, while a long time

frame is used for forecasting long-term price movements. There are four types prices that are important:

(1) Open: it is the price at the beginning of the time frame.

(2) Close: it is the price at the closing of the time frame.

(3) High: it is the highest price in the time frame.

(4) Low: it is the lowest price in the time frame.

Besides the trading price, trading volume is important as well. There are two types of volume: buy volume and sell volume. This trading information is used to plot the trading charts to be used in technical analysis. There are several types of charts:

(1) Line Chart. It uses only closing price to plot along with time period.

Figure 2.1 Line Chart. (What Are Charts?, Stockcharts.com)

(2) Bar Chart. The high, low and closing prices are used to plot a bar chart. The high and low are represented by the top and bottom of the vertical bar and the close is the short horizontal line crossing the vertical bar. It can also include the open into the chart as well.

Figure 2.2 Bar Chart with high, low, and closing price. (What Are Charts?, Stockcharts.com)

(3) Candlestick Chart. Originating in Japan over 300 years ago. It includes all the high, low, open and closing prices to form a chart. It has become quite popular in recent years. There are two types of candlesticks: white (clear) candlestick and black (solid) candlestick. A white candlestick is formed when the open is lower than the closing, while a black candlestick is formed when the open is higher than the closing.

(4) Point & Figure Chart. This chart is based solely on price movement, and does not take time into consideration, unlike the above charts that still plot even when there is no price movement. In a point & figure chart, little or

Figure 2.3 Bar Chart with high, low, open, and closing price. (What Are Charts?, Stockcharts.com)

no price movement is not plotted on the chart, so there is no duplication on the chart. Only price movements that exceed specified levels are plotted.

From these charts, many price movement patterns, overlays, and indicators have been invented to analyze and predict the stock's price. One of the simplest and most popular among them is the **moving average**.

A moving average is calculated by averaging the price over the past n days. Usually, it is calculated from closing prices. Its main purpose is to identify the direction of the trend. The two most popular types of moving averages are the following:

(1) Simple Moving Average (SMA). It is formed by computing the average price of a stock over a specific time period. Usually, it is based on the closing price. An n-day simple moving average is the n days sum of closing prices divided by n.

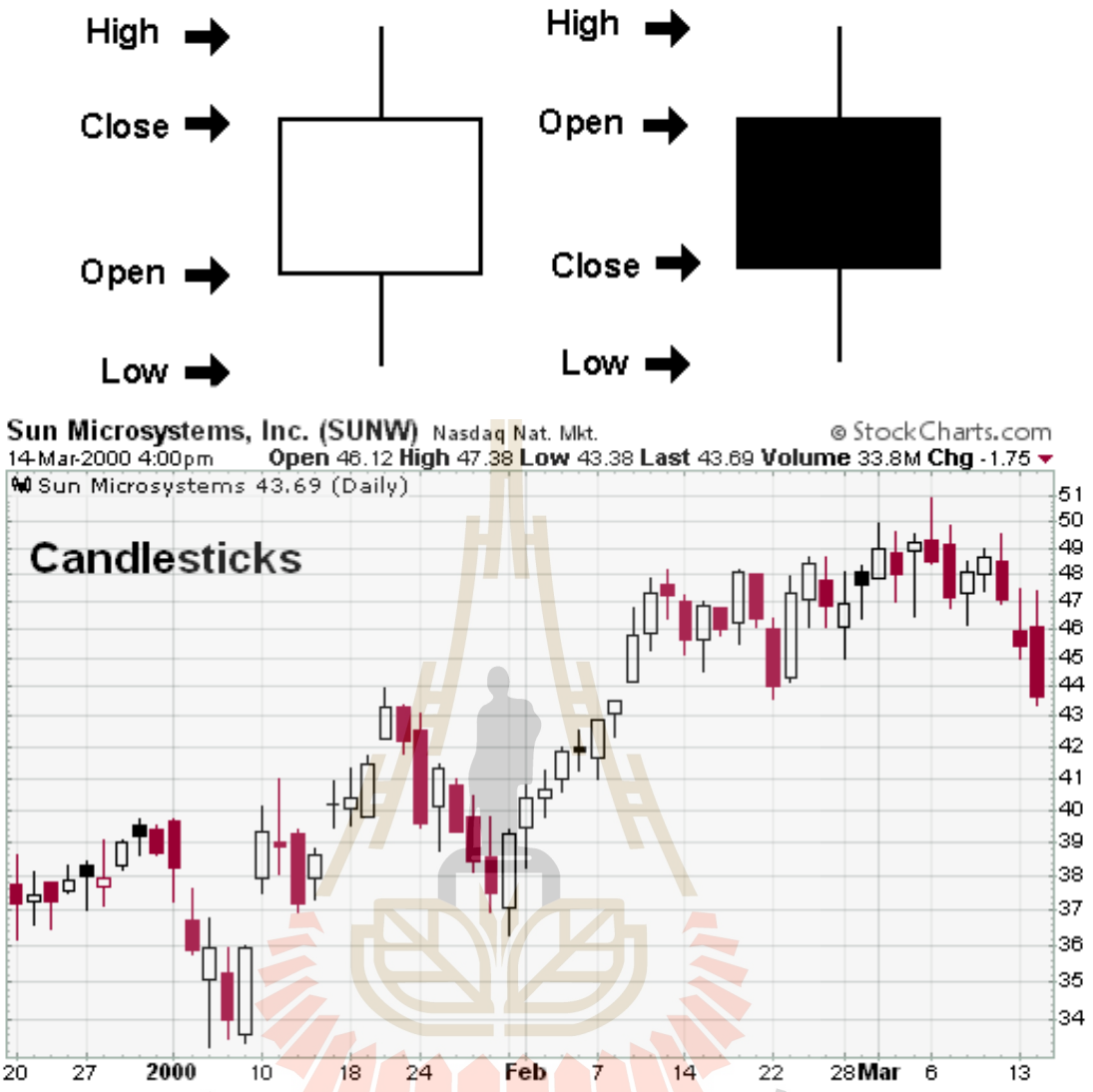

Figure 2.4 Candllestick Chart. (What Are Charts?, Stockcharts.com)

(2) Exponential Moving Average (EMA). It reduces the lag in SMA by applying more weight to recent prices. The weighting applied to the most recent price depends on the number of periods in the moving average. There are three steps in calculating the n days EMA:

- SMA: n period sum divided by n. It is used as the first EMA.
- Multiplier: $\frac{2}{n+1}$.
- EMA: (Close - EMA(previous day)) $\times$ Multiplier + EMA(previous day).

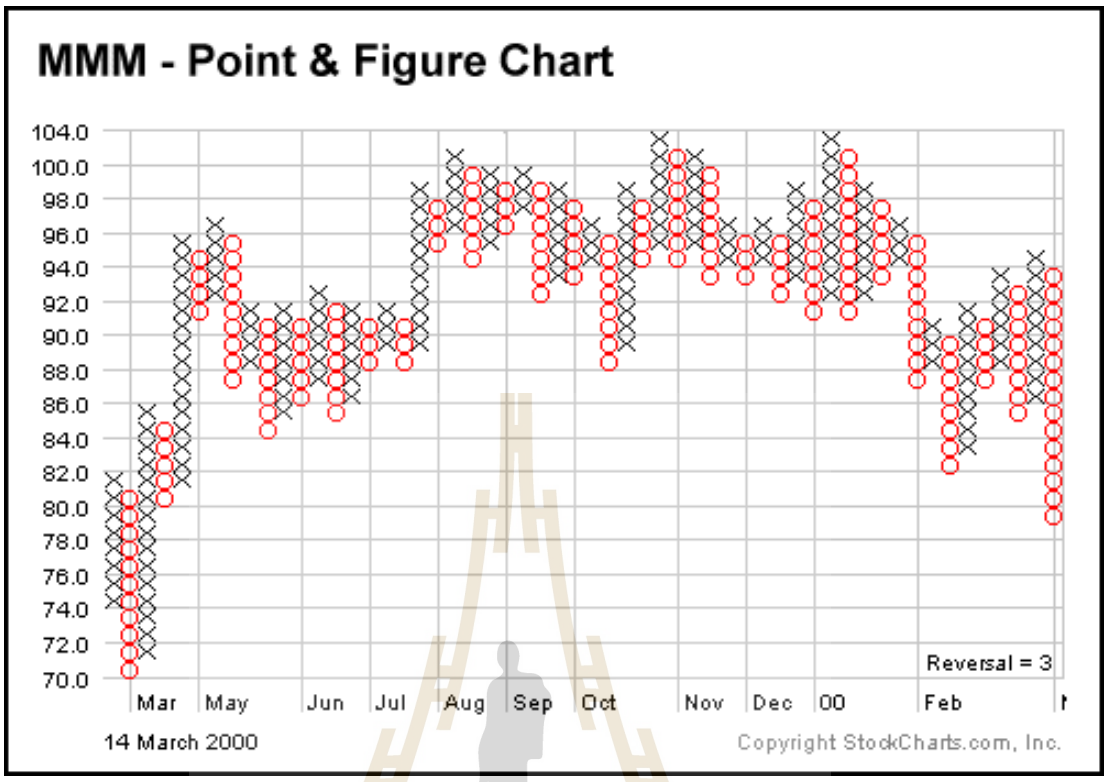

Figure 2.5 Point & Figure Chart. (What Are Charts?, Stockcharts.com)

Figure 2.6 SMA & EMA. (Moving Averages - Simple and Exponential, Stockcharts.com)

Moving average convergence-divergence (MACD) is one of the simplest and most effective indicators developed by Gerald Appel. It is based on moving average. Generally, it is formed by 12-day and 26-day EMAs.

1. MACD Line: (12-day EMA) - (26-day EMA)

2. Signal Line: 9-day EMA of MACD Line

3. MACD Histogram: MACD Line - Signal Line

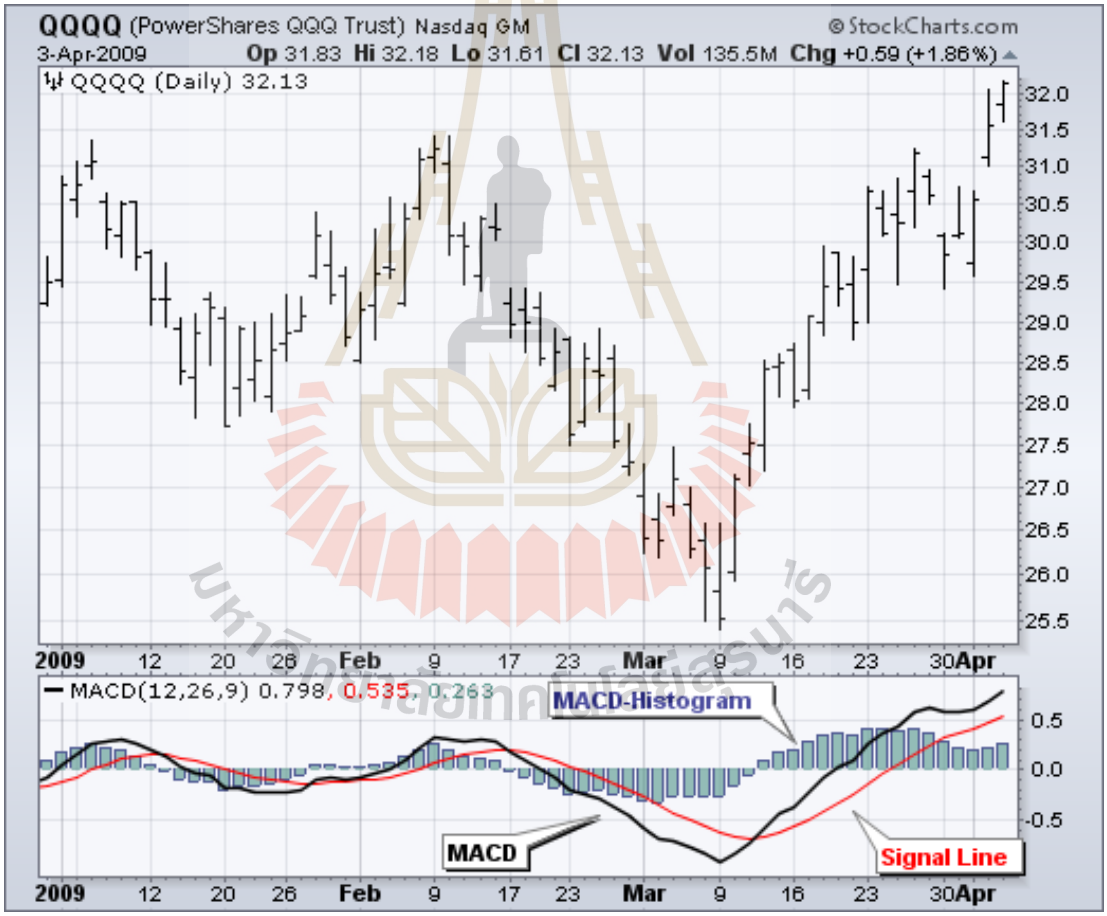

Figure 2.7 MACD. (Moving Average Convergence-Divergence (MACD), Stockcharts.com)

There are many technical tools that have been developed in this area. Their usages and purposes are varied. One tool may be useful in some situation and may

not in another. Generally, investors apply several tools to help them in identifying the right time to buy and sell the stocks.

2.4 Support Vector Machine

A Support Vector Machine (SVM) is one of the popular approaches in machine learning. (Fletcher, 2009). It is used mainly in classification tasks and regression analysis. Its basic concept is to divide a set of data into two groups by drawing a separation boundary between two sets of data. It is supervised learning, which means we need to present some sets of training data along with classified results in order to allow the SVM to determine an appropriated separation function. After it has been trained with the training data, it is used to predict which group the input data belongs to. This makes SVM a non-probabilistic binary linear classifier.

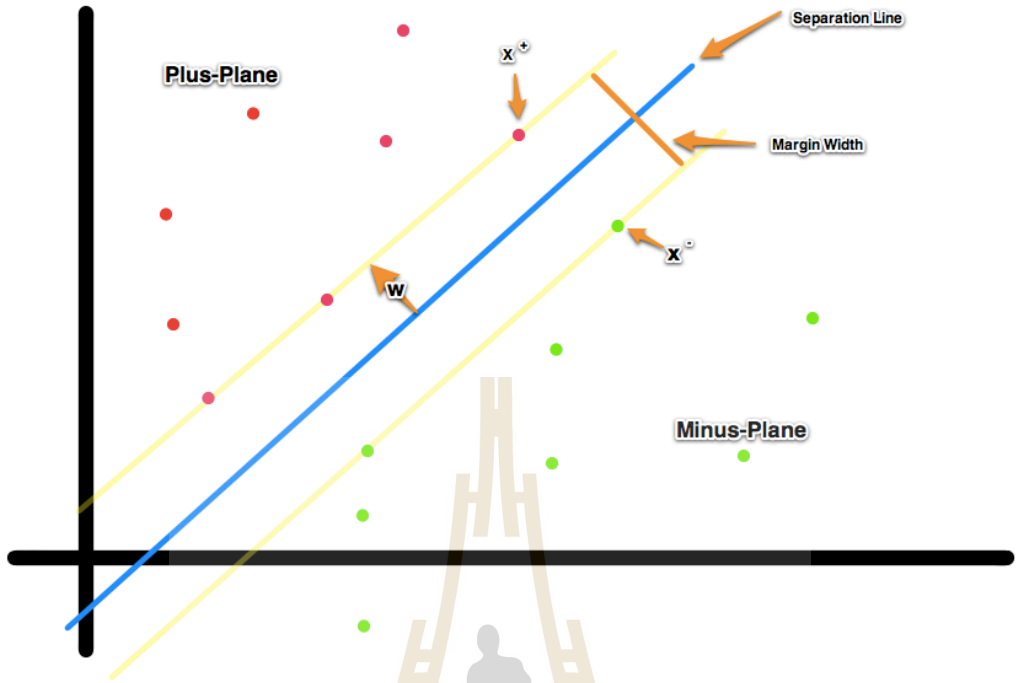

Figure 2.8 Linear SVM.

- Input Data: $x_i \in \mathbb{R}^n, i = 1, ..., L$
- Output Data: $y_i \in \{-1, 1\}, i = 1, ..., L$
- $w \in \mathbb{R}^n, b \in \mathbb{R}$
- Plus-plane: $x_i \cdot w + b \geq 1, y_i = 1$
- Minus-plane: $x_i \cdot w + b \leq -1, y_i = -1$

There are many situations where a straight line cannot separate the data, i.e., the data is not linearly separable. In these cases, SVM will first map the data from the input space into the feature space (which usually has a higher dimension than the input space) by using a Kernel function, and then do all the work in this new feature space instead (where the data is now linearly separable). So the key is to find a suitable Kernel function to be applied. There are many Kernel functions that have been constructed until now. We will come back to this Kernel

topic later.

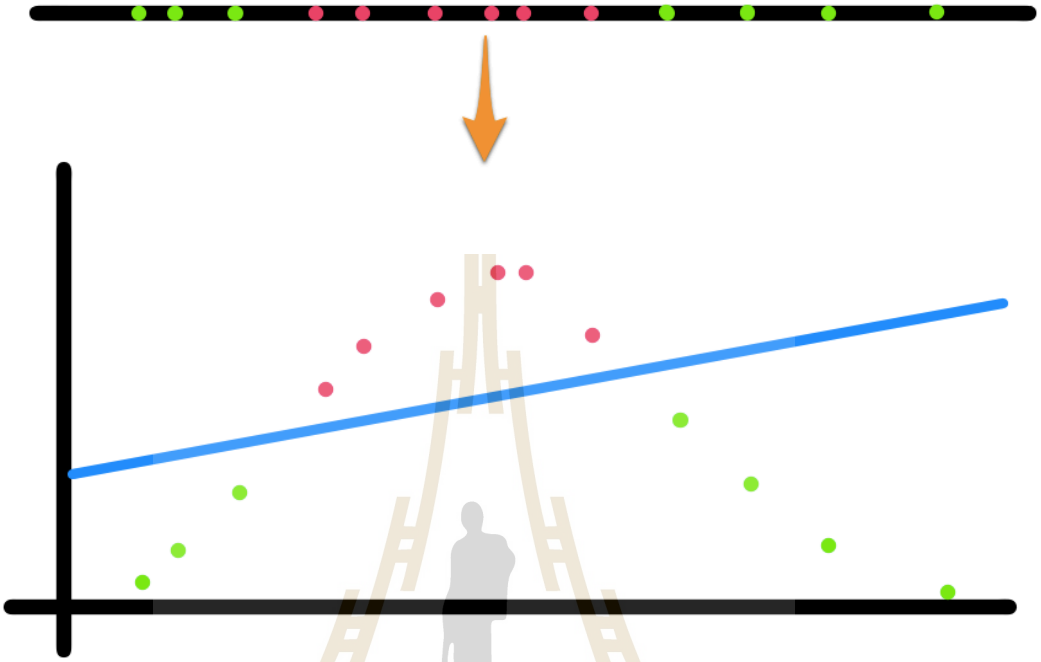

Figure 2.9 Nonlinear SVM.

In case that data is almost linearly separable, i.e., there are some points, which are not linearly separable; we can overcome this by allowing some error in the equations, $\xi_i, i = 1, ..., L$. This is called **soft margin model**.

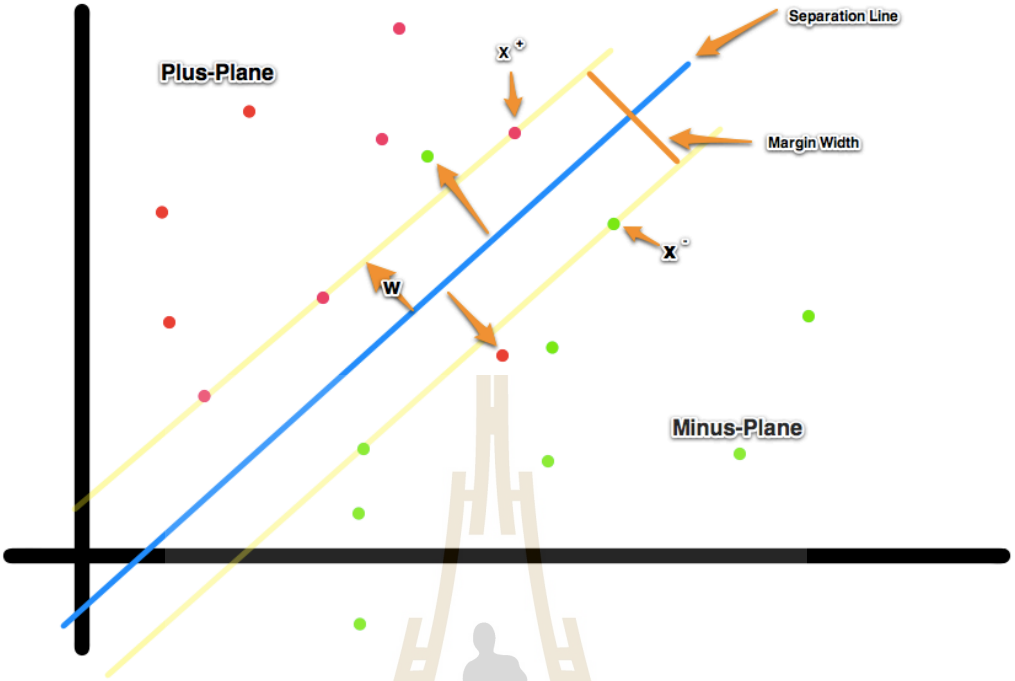

Figure 2.10 Soft Margin SVM.

- Input Data: $x_i \in \mathbb{R}^n, i = 1, ..., L$
- Output Data: $y_i \in \{-1, 1\}, i = 1, ..., L$
- $w \in \mathbb{R}^n, b \in \mathbb{R}$
- Plus-plane: $x_i \cdot w + b \geq 1 - \xi_i, y_i = 1$
- Minus-plane: $x_i \cdot w + b \leq -1 + \xi_i, y_i = -1$

Originally, a support vector machine was designed to classify data into two classes. By combining with some comparison technique, we can expand it to support multi-classes classification. This is called **multi-classes SVM**. The commonly used methods are the following:

(1) One-versus-all. It compares one class and the rest, and repeats the process until the data is assigned into any class.

(2) One-versus-one. It compares every possible pair of classes, and then the

class with the highest vote is the assigned class.

The concept of support vector can be extended to predict the real number data. Instead of using support vectors used to separate data into groups, support vectors are used as the possible range of value. This is called **support vector regression** (SVR).

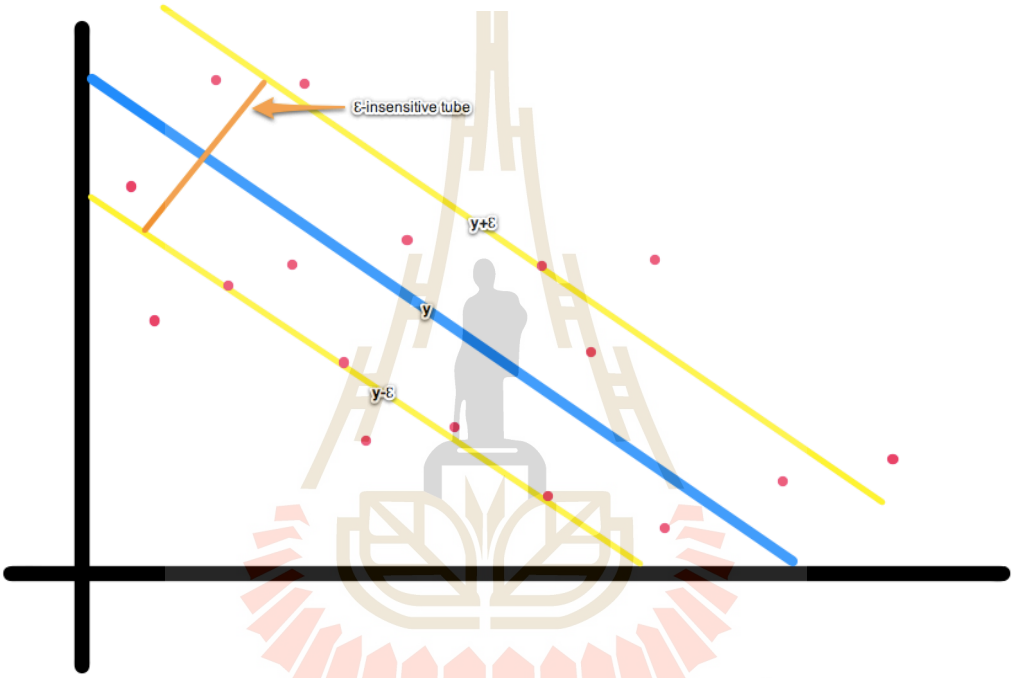

Figure 2.11 SVR.

- Input Data: $x_i \in \mathbb{R}^n, i = 1, ..., L$
- Output Data: $y_i \in \mathbb{R}, i = 1, ..., L$
- $w \in \mathbb{R}^n, b \in \mathbb{R}$
- Predicted Equation: $y_i = w \cdot x_i + b$
- Actual Value: $t_i \in \mathbb{R}, i = 1, ..., L$
- ϵ-insensitive tube: $|t_i - y_i| < \epsilon$

2.4.1 Kernel Trick

Kernel trick is a methodology to map the data from an input space S into an inner product space V, without having to compute the mapping explicitly. (Gunn, 1998) The trick to avoid the explicit mapping is to use learning algorithms that only requires dot products between the vectors in V, and choose the mapping such that these high-dimensional dot products can be computed within the original space, by **kernel function**.

Let $K(x, y)$ be kernel function defined as:

$$K(x, y) = \langle \varphi(x), \varphi(y) \rangle_v \tag{2.1}$$

where

$S \subseteq \mathbb{R}^n, V \subseteq \mathbb{R}^m$ such that $m, n \in \mathbb{R}, m > n$

$\langle \cdot, \cdot \rangle_v$ is an inner product (dot product) in V

$\varphi : S \to V$

There are many kernel functions, which are suitable for different tasks.

(1) Linear Function. A linear mapping is the most simplest kernel function. It is defined as:

$$K(x, y) = \langle x, y \rangle \tag{2.2}$$

(2) Polynomial. A polynomial mapping is a popular method for non-linear modeling. It is defined in a form:

$$K(x, y) = (\rho \langle x, y \rangle + \varrho)^d \tag{2.3}$$

for a certain values of the scale ρ, and offset ϱ.

(3) Gaussian Radial Basis Function. Radial basis functions (RBF) have received significant attention, most commonly with a Gaussian of the form:

$$K(x, y) = exp\left(-\sigma||x - y||^2\right) \tag{2.4}$$

(4) Laplace Radial Basis Function. A radial basis function (RBF) of the form:

$$K(x, y) = exp(-\sigma||x - y||) \tag{2.5}$$

(5) Bessel Kernel.

$$K(x, y) = Bessel^n_{\nu+1}(\sigma|||x - y||^2) \tag{2.6}$$

where $Bessel$ is the Bessel function of first kind.

(6) Hyperbolic Tangent Kernel. It is also know as the Sigmoid Kernel or Multilayer Perceptron (MLP).

$$K(x, y) = tanh(\rho\langle x, y\rangle + \varrho) \tag{2.7}$$

for certain values of the scale ρ, and offset ϱ.

(7) ANOVA RBF Kernel. It is also a variance of radial basis function (RBF).

$$K(x, y) = \sum_{1\ leqi_1 \ldots \leq i_D \leq N} \Pi^D_{d=1} K_G(x_{i_d}, y_{i_d}) \tag{2.8}$$

where $K_G(x, y)$ is Gaussian RBF Kernel.

(8) Splines. Splines are a popular choice for modeling due to their flexibility. A

finite spline, of order M, with N knots located at τ_S is given by:

$$K(x,y) = \sum_{r=0}^{M} x^r \cdot y^r + \sum_{s=1}^{N} (x-\tau_s)_+^M \cdot (y-\tau_s)_+^M \tag{2.9}$$

An infinite spline is defined on the interval $[0,1)$ by:

$$K(x,y) = \sum_{r=0}^{M} x^r \cdot y^r + \int_0^1 (x-\tau_s)_+^M \cdot (y-\tau_s)_+^M d\tau \tag{2.10}$$

In the case when $M = 1$, (S_1^∞), the kernel is given by:

$$K(x,y) = 1 + \langle x,y \rangle + \frac{1}{2}\langle x,y \rangle min(x,y) - \frac{1}{6} min(x,y)^3 \tag{2.11}$$

where the solution is a piece-wise cubic.

The linear kernel is usually used in the area of text categorization, while the polynomial kernel is popular in the field of image processing. The Gaussian RBF, Laplace RBF, and Bessel kernels are general purpose kernels for using in case there is no prior knowledge of the dataset. (Karatzoglou et al., 2006).

2.4.2 Empirical Mode Decomposition

"The Fourier analysis has been used as a key concept in linear analysis since the late of 18th century. The disadvantage of Fourier transform (FT) family except short-term Fourier transform (STFT) is in the representation of amplitude and frequency plane, meaning not to be able to handle nonlinear functions. Huang et al. developed an a posteriori algorithm for analysing nonlinear and nonstationary datasets using EMD with adaptive control over a separate data structure, and it was known as the Hilbert-Huang transform (HHT). This new transform overcame the limitation of Hilbert transform, which was only suitable for a narrow band-

passed signal. EMD is based on the local time scale of the signal. Because the decomposition is based on the local characteristic time scale of the data, it is applicable to nonlinear and nonstationary processes." (Premanode, 2013)

The Empirical Mode Decomposition (EMD) is a decomposition method used to transform any given data into Intrinsic Mode Functions (IMF). (Huang and Shen, 2005) An IMF is defined as a function with the following properties:

(1) The number of extrema and the number of zero-crossings must either be equal or differ at most by one.

(2) At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

Originally, an EMD was proposed as a part of the Hilbert-Huang transform (HHT). The IMF extracting procedure is called sifting. It works as follows:

(1) Identify all the local extrema in the data.

(2) Use cubic splines to produce the upper envelope from all of the maxima identified in step (1).

(3) Repeat step (2) to produce the lower envelope from the minima.

(4) Calculate mean of the upper and lower envelopes, m_1.

(5) Subtract the mean from the data

$$X(t) - m_1 = h_1 \tag{2.12}$$

(6) Repeat step (1) - (5) with the residue, h_1, as the data.

These processes can be illustated as a flowchart in figure 2.12.

Traditionally, there are two stoppage criteria used to determine the number of the sifting steps:

(1) The first criterion is similar to the Cauchy convergence test, we define a sum of the difference, SD, as

$$SD_k = \frac{\sum_{t=0}^{T} |h_{k-1}(t) - h_k(t)|^2}{\sum_{t=0}^{T} h_{k-1}^2(t)} \tag{2.13}$$

Then we repeat the sifting process until SD is smaller than a pre-determined value.

(2) A second criterion is based on a predefined number, called S-number. The sifting process will stop when for S consecutive times of the sifting, the numbers of zero-crossings and extrema stay the same, and are equal or at most differ by one.

Once the sifting process meets the stoppage criterion, we get the first IMF, c_1. Then we subtract c_1 from the data, $X(t) - c_1 = r_1$, and repeat the process with r_1 as the data.

The sifting process stops when the residue, r_n, becomes a monotonic function from which no more IMF can be extracted. So we get:

$$X(t) = \sum_{j=1}^{n} c_j + r_n \tag{2.14}$$

In the next chapter, we propose a new transformation for turning a non-stationary stock data into a stationary dataset for using in the forecasting model. The prediction model and the performance measurement used in this study are presented.

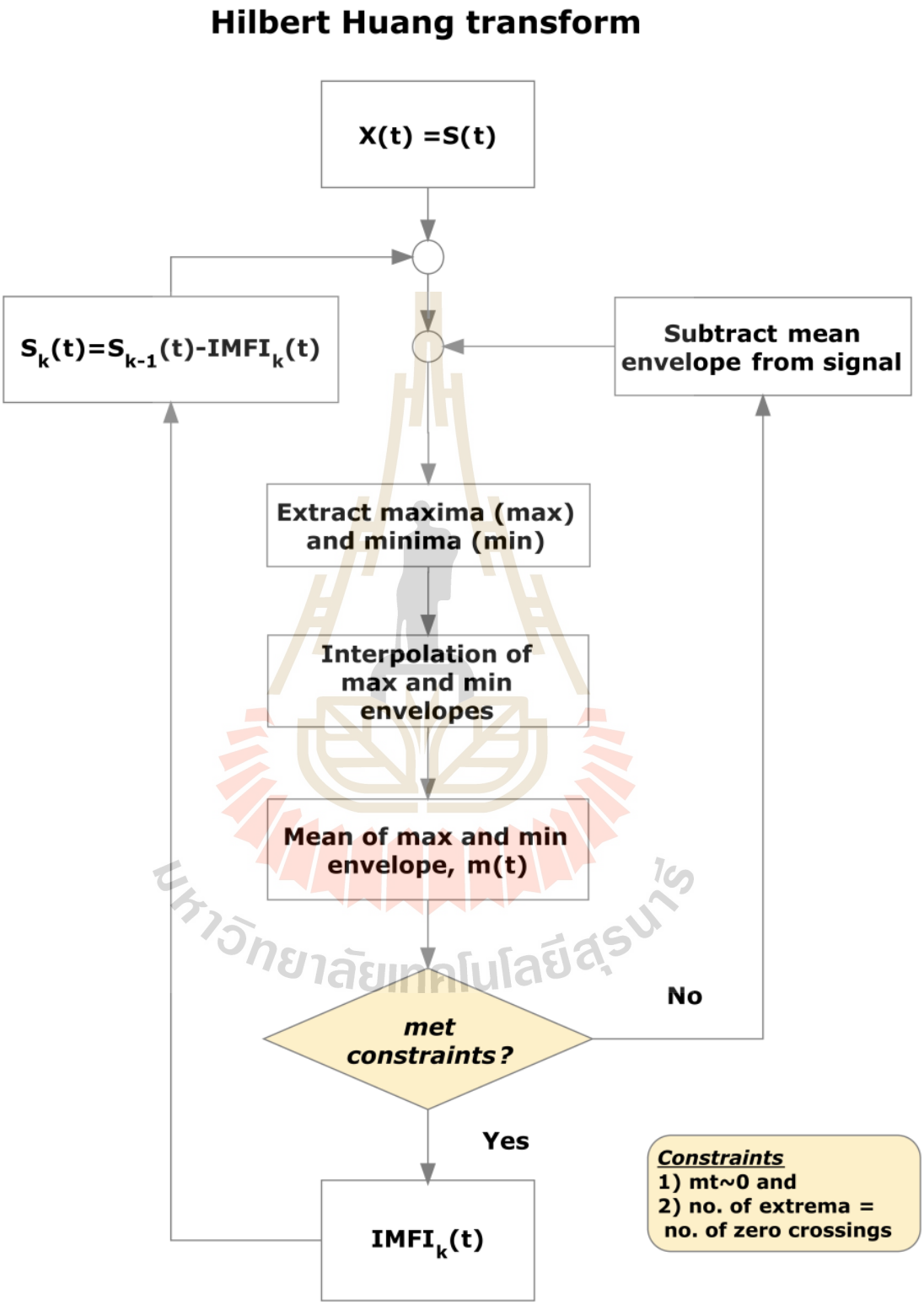

Figure 2.12 The sifting process in EMD. (Premanode, 2013)

CHAPTER III

METHODOLOGY

In this section, we are going to discuss the mechanism used in this study. The data that we used to train and test is the closing price on daily basis. We used R programming with Kernlab library in our experiments.

At any given period, there are four points of stock price of interest, i.e., open, close, high, low. Usually, the closing price is used to represent a stock price at any given time. Therefore, we use the closing price as a base of stock data for training and prediction in this thesis.

Generally, a closing price is non-stationary data, which is not suitable for forecasting (Iordanova, 2009). Therefore, we are going to transform this closing price into stationary data before feeding into the prediction models. The closing price difference, e.g., $x_i - x_{i-1}$, is a stationary series; we call this *closing change*. One disadvantage of this calculation is the loss of one observation.

Although the level of demand and supply in the stock market determines stock price, its price cannot vary freely, i.e., there is a minimum amount by which a stock price can change, called *tick size* or *price spread*. In SET, tick size varies with different price intervals, e.g., tick size is 0.01 Baht for prices between 0.01 Baht and 2.00 Baht. This results in changing of range of closing change for stock whose price changes from one interval to another interval, which may result in variance changing over time. We calculate the number of ticks in closing change; we call this *tick change*, which is stationary, and its range remain the same for any price interval.

Table 3.1 The current price spread in effect at the time of writing this thesis.

Market Price Level (Baht)	Spread (Baht)
Less than 2	0.01
2 up to less than 5	0.02
5 up to less than 10	0.05
10 up to less than 25	0.10
25 up to less than 100	0.25
100 up to less than 200	0.50
200 up to less than 400	1.00
400 up	2.00

Let

$close_i$ be closing price of i^{th} day

$tickCount_i$ be the number of tick of closing price of i^{th} day

$change_i$ be the closing change of i^{th} day

$tickChange_i$ be the tick change of i^{th} day

$[a_i, a_{i+1}]$ be the i^{th} price interval

t_i be the tick size of an i^{th} price interval

$$change_i = close_i - close_{i-1}$$

$$tickCount_i = \sum_{j=0}^{n} \frac{a_{j+1} - a_j}{t_j} + \frac{close_i - a_{n+1}}{t_{n+1}}; close_i \in [a_{n+1}, a_{n+2}]$$

$$tickChange_i = tickCount_i - tickCount_{i-1}$$

Figure 3.1 illustrates charts from closing price, closing change, and tick change from three stocks. The horizontal line in the closing price charts indicates the point where tick size is changed. For stock (A), the tick size remains the same for the entire sample period, so the closing change chart and tick change chart are very similar to each another. Stock (B) has three different tick sizes in the sample data, but the majority of its data is in a single tick size interval. So its closing

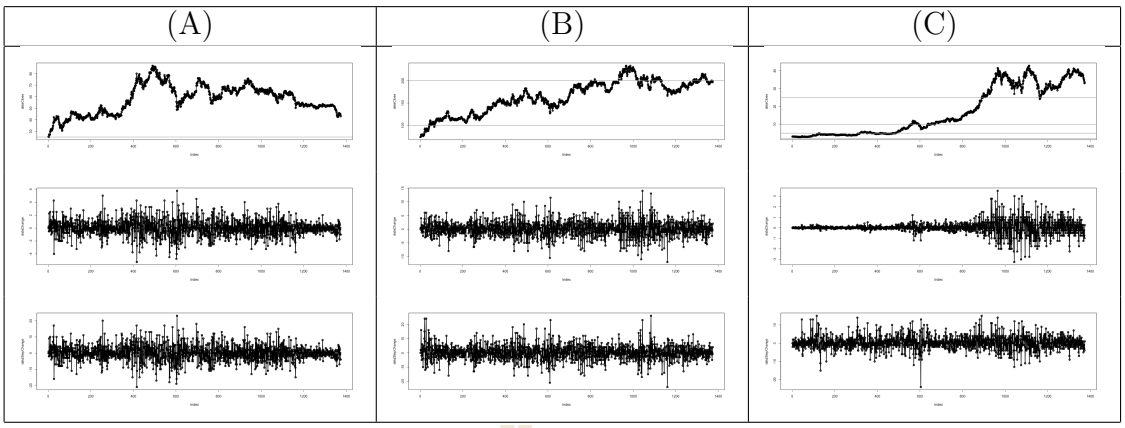

Figure 3.1 Chart of closing price (top), closing change (middle), and tick change (bottom) from three stocks in SET50.

change and tick change charts are quite similar to one another. Stock (C) has four different tick size intervals in the dataset, and each tick size interval has a significant amount of data. So its closing change and tick change charts are quite different from one another, and its variance is likely to be changed overtime when tick size is changed.

We used ADF (Augmented Dickey-Fuller) and KPSS (Kwiatkowski–Phillips–Schmidt–Shin) tests for testing if the data is stationary. The null-hypothesis for an ADF test is that the data are non-stationary. So large p-values are indicative of non-stationarity, and small p-values suggest stationarity. Usually, a threshold of 5% is used, i.e., a p-value greater than 0.05 suggests non-stationary of the data. For KPSS test, this reverses the hypotheses, so the null-hypothesis is that the data are stationary. In this case, small p-values, e.g., less than 0.05, suggest that data is non-stationary. Tables 3.2, 3.3, and 3.4 show the test results for closing price, closing change, and tick change repectively. The data used for these stationary tests is from January, 1991 to August 15, 2016, except for some stock that was added to the SET50 after January, 1991. From the results, the closing price of all stock fails one or both of stationary tests. For closing change, it passes one or both tests for all stock. There are eight stocks that fail the KPSS

Table 3.2 Stationary test result for closing price.

Stock	ADF	KPSS		Stock	ADF	KPSS
ADVANC	Fail	Fail		IVL	Fail	Fail
AOT	Fail	Fail		KBANK	Fail	Fail
BA	Fail	Fail		KCE	Fail	Fail
BANPU	Fail	Fail		KTB	Fail	Fail
BBL	Fail	Fail		LH	Fail	Fail
BCP	Pass	Fail		MINT	Fail	Fail
BDMS	Fail	Fail		MTLS	Pass	Fail
BEC	Fail	Fail		PS	Fail	Fail
BH	Fail	Fail		PTT	Fail	Fail
BLA	Fail	Fail		PTTEP	Fail	Fail
BTS	Pass	Fail		PTTGC	Fail	Fail
CBG	Fail	Fail		ROBINS	Fail	Fail
CENTEL	Fail	Fail		SAWAD	Fail	Fail
CK	Fail	Fail		SCB	Fail	Fail
CPALL	Fail	Fail		SCC	Fail	Fail
CPF	Fail	Fail		TASCO	Fail	Fail
CPN	Fail	Fail		TCAP	Fail	Fail
DELTA	Fail	Fail		TMB	Fail	Fail
DTAC	Fail	Fail		TOP	Fail	Fail
EGCO	Fail	Fail		TPIPL	Fail	Fail
GLOW	Fail	Fail		TRUE	Fail	Fail
GPSC	Fail	Fail		TTW	Fail	Fail
HMPRO	Fail	Fail		TU	Fail	Fail
INTUCH	Fail	Fail		WHA	Fail	Fail
IRPC	Pass	Fail		-	-	-

Table 3.3 Stationary test result for closing change.

Stock	ADF	KPSS		Stock	ADF	KPSS
ADVANC	Pass	Pass		IVL	Pass	Pass
AOT	Pass	Fail		KBANK	Pass	Pass
BA	Pass	Pass		KCE	Pass	Fail
BANPU	Pass	Pass		KTB	Pass	Pass
BBL	Pass	Pass		LH	Pass	Pass
BCP	Pass	Fail		MINT	Pass	Fail
BDMS	Pass	Fail		MTLS	Pass	Pass
BEC	Pass	Pass		PS	Pass	Pass
BH	Pass	Fail		PTT	Pass	Pass
BLA	Pass	Pass		PTTEP	Pass	Pass
BTS	Pass	Fail		PTTGC	Pass	Pass
CBG	Pass	Pass		ROBINS	Pass	Fail
CENTEL	Pass	Pass		SAWAD	Pass	Pass
CK	Pass	Pass		SCB	Pass	Pass
CPALL	Pass	Pass		SCC	Pass	Pass
CPF	Pass	Pass		TASCO	Pass	Pass
CPN	Pass	Pass		TCAP	Pass	Pass
DELTA	Pass	Pass		TMB	Pass	Pass
DTAC	Pass	Pass		TOP	Pass	Pass
EGCO	Pass	Pass		TPIPL	Pass	Pass
GLOW	Pass	Pass		TRUE	Pass	Pass
GPSC	Pass	Pass		TTW	Pass	Pass
HMPRO	Pass	Pass		TU	Pass	Pass
INTUCH	Pass	Pass		WHA	Pass	Pass
IRPC	Pass	Pass		-	-	-

Table 3.4 Stationary test result for tick change.

Stock	ADF	KPSS		Stock	ADF	KPSS
ADVANC	Pass	Pass		IVL	Pass	Pass
AOT	Pass	Fail		KBANK	Pass	Pass
BA	Pass	Pass		KCE	Pass	Pass
BANPU	Pass	Pass		KTB	Pass	Pass
BBL	Pass	Pass		LH	Pass	Pass
BCP	Pass	Pass		MINT	Pass	Pass
BDMS	Pass	Pass		MTLS	Pass	Pass
BEC	Pass	Pass		PS	Pass	Pass
BH	Pass	Pass		PTT	Pass	Pass
BLA	Pass	Pass		PTTEP	Pass	Pass
BTS	Pass	Pass		PTTGC	Pass	Pass
CBG	Pass	Pass		ROBINS	Pass	Pass
CENTEL	Pass	Fail		SAWAD	Pass	Pass
CK	Pass	Pass		SCB	Pass	Pass
CPALL	Pass	Pass		SCC	Pass	Pass
CPF	Pass	Pass		TASCO	Pass	Pass
CPN	Pass	Pass		TCAP	Pass	Pass
DELTA	Pass	Pass		TMB	Pass	Pass
DTAC	Pass	Pass		TOP	Pass	Pass
EGCO	Pass	Pass		TPIPL	Pass	Pass
GLOW	Pass	Pass		TRUE	Pass	Pass
GPSC	Pass	Pass		TTW	Pass	Pass
HMPRO	Pass	Pass		TU	Pass	Pass
INTUCH	Pass	Pass		WHA	Pass	Pass
IRPC	Pass	Pass		-	-	-

test. For tick change, it gives better results than closing change. All stock pass the ADF test, and there are only two stocks that fail the KPSS test. This shows that the tick change transformation is more suitable to transform stock price into stationary data than closing change.

We used closing price, closing change, and tick change for the experiment in this study. Furthermore, we only uses historical data of individual stock price alone for forecasting its price without any other data, because we need to determine the relation between each data, which may result in different sets of related data for difference stock.

With Kernlab package in R programming that we used in this study, there are three implementations for SVR, e.g., Nu-SVR, EPS-SVR, and EPS-BSVR. Each implementation calculates the error function slightly differently, which should not make any significant impact to the result.

For kernel function selection, Gaussian RBF is a preferred choice of kernel for financial forecasting (Tay and Cao, 2001; Chen and Shih, 2006; Kumar and Thenmozhi, 2009; Huang, Nakamori, and Wang, 2005). In addition, Gaussian RBF, Laplace RBF, and Bessel kernels are suitable for datasets with no prior knowledge (Karatzoglou et al., 2006). So we used Gaussian RBF, Laplace RBF, and Bessel kernels for our experiment. Since we can recursively use its result to forecast further, e.g., use tomorrow's predicted result to predict the day after tomorrow, we therefore focus on prediction for a single time period ahead, i.e., one day ahead.

The error was measured by Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), and R-squared (R^2). MAPE measures an error in percentage. It is commonly used in quantitative forecasting methods (Sutheebanjard and Premchaiswadi, 2010). MSE measures the average of the squares of the

errors. R-squared is a number that indicates how well the data fit the prediction model.

$$MAPE = \frac{\sum_{i=1}^{n} \left| \frac{a_i - p_i}{a_i} \right|}{n} \times 100$$

$$MSE = \frac{\sum_{i=1}^{n} (a_i - p_i)^2}{n}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (a_i - p_i)^2}{\sum_{i=1}^{n} (a_i - \bar{a})^2}$$

where a_i is an actual value.

p_i is a predicted values.

n_i is a number of data.

The next chapter gives a detail and discussion about the data used in this study.

CHAPTER IV

THE DATA

The data used for experiments in this thesis were the closing price data of stocks in SET50, and retrieved from E-Finance on August 16, 2016. The data was from January 2, 2014 to August 15, 2016, except for some stock that was added to SET50 after January 2, 2014. Therefore, the number of records is varied, ranging from around 200 records to 600 records depending on the stock. We used 70% of data for training, and 30% for validation.

Table 4.1 shows the stocks and the number of records that we used in this study. For BEM, it was placed into the SET50 at the begining of 2016, so there are not enough records for use in our test because some of our forecasting models require a substantial number of records. So BEM is not included in the test. Therefore, the number of stock we used in our test is 49 stocks.

Table 4.2 shows a sample of stock data, e.g., closing price, closing change and tick change.

In addition to using closing change, and tick change as basis for prediction model, we used them to create an indicator for buying-selling signal for stocks similar to the indicators from technical analysis. We create a simple indicator from moving average of closing change, and tick change, i.e., simple moving average, and exponential moving average. We buy stock when this indicator crosses above the zero line, and sell when it crosses below the zero line. For comparison, we also

Table 4.1 Stocks used for training and testing in this study.

Stock	Records		Stock	Records
ADVANC	636		IVL	636
AOT	636		KBANK	636
BA	183		KCE	636
BANPU	636		KTB	636
BBL	636		LH	636
BCP	636		MINT	636
BDMS	636		MTLS	165
BEC	636		PS	636
BH	636		PTT	636
BLA	636		PTTEP	636
BTS	636		PTTGC	636
CBG	168		ROBINS	636
CENTEL	636		SAWAD	303
CK	636		SCB	636
CPALL	636		SCC	636
CPF	636		TASCO	636
CPN	636		TCAP	636
DELTA	636		TMB	636
DTAC	636		TOP	636
EGCO	636		TPIPL	636
GLOW	636		TRUE	636
GPSC	55		TTW	636
HMPRO	636		TU	636
INTUCH	636		WHA	636
IRPC	636		BEM	-

did a simulation using the popular MACD signal as a reference.

SMA_i is SMA of i^{th} day

EMA_i is EMA of i^{th} day

$close_i$ is a closing price of i^{th} day

$$\text{n-day SMA} : SMA_m = \frac{\sum_{i=0}^{n-1} close_{m-i}}{n}$$

$$\text{n-day EMA} : EMA_m = (close_m - EMA_{m-1}) \times \frac{2}{n+1} + EMA_{m-1}$$

We simulated transaction, e.g., buying-selling stock, from the signal of the

Table 4.2 The sample data used for training and testing from CENTEL.

Closing Price	Closing Change	Tick Change
23.2	0.4	4
23.1	-0.1	-1
23.3	0.2	2
24.8	1.5	15
24.7	-0.1	-1
25.25	0.55	4
25.75	0.5	2
25.25	-0.5	-2
24.9	-0.35	-2
25	0.1	1
25.5	0.5	2
27	1.5	6
27.25	0.25	1

above indicators. For each transaction, we buy and sell a single stock and calculate a percentage of profit (or loss) from that transaction. We did simulation from 2 to 20 days SMA, and EMA of closing change, and tick change. Table 4.3, 4.4, 4.5, 4.6, and 4.7 only show the results from 20-day SMA, and EMA, since it gave the best results for our datasets.

Table 4.3 Buy-sell simulation result from MACD.

Stock	Transactions	Profit (%)	Stock	Transactions	Profit (%)
ADVANC	27	-55.121744939526	IVL	25	-16.185719078391
AOT	23	43.9534924062185	KBANK	24	-7.85228079674393
BA	7	7.67018263743849	KCE	28	58.3229147146618
BANPU	24	-45.333031060801	KTB	24	-2.15053763440859
BBL	34	-38.9823373062835	LH	27	-28.443642811061
BCP	30	-21.0315060752058	MINT	25	34.312228579352
BDMS	19	46.7812057422813	MTLS	5	-4.38271779717489
BEC	29	-49.4296409563456	PS	25	2.01823787597696
BH	29	-7.17237728697927	PTT	21	11.3842842933692
BLA	22	-29.0801619505111	PTTEP	20	-20.2673798028441
BTS	28	-0.415184879067223	PTTGC	28	-33.9941476888655
CBG	7	27.6708860756313	ROBINS	25	-13.1381131760876
CENTEL	26	-4.57920026140498	SAWAD	11	-18.9609355166952
CK	20	48.4602060100471	SCB	24	0.149067504545156
CPALL	25	8.25525347167562	SCC	21	10.6209091282463
CPF	26	-17.4016434835999	TASCO	24	-0.625000000000005
CPN	27	-13.6643965449889	TCAP	23	21.2576501689121
DELTA	27	-22.4793418802553	TMB	28	-19.0130626419783
DTAC	19	-26.0513007962265	TOP	29	-34.5737266897063
EGCO	32	-5.02260705227328	TPIPL	19	19.1596799710994
GLOW	34	-49.545721219475	TRUE	23	17.0261945274707
GPSC	3	0.859138742561701	TTW	28	-8.18804014805957
HMPRO	22	2.27882037533512	TU	29	-14.1106045329053
INTUCH	26	-3.532462147423	WHA	24	3.29162113375672
IRPC	21	33.8162367020105	BEM	-	-

Table 4.4 Buy-sell simulation result from SMA of closing price.

Stock	Transactions	Profit (%)	Stock	Transactions	Profit (%)
ADVANC	25	-6.89380256763199	IVL	27	20.0254298535954
AOT	29	33.1786584104143	KBANK	28	1.86835601118035
BA	5	-1.12420495014176	KCE	21	69.1303538765374
BANPU	26	7.27944108662104	KTB	27	-5.9689192037344
BBL	36	-7.32377017385525	LH	24	-13.5314932979729
BCP	31	-9.67478688109438	MINT	38	16.3732379174924
BDMS	29	24.8640234172919	MTLS	7	-10.1281626503367
BEC	32	-56.3608189159935	PS	27	16.6596830258591
BH	25	57.7517373373587	PTT	27	20.1468832943797
BLA	30	-16.5397148247004	PTTEP	31	3.68695837841823
BTS	19	10.181644793919	PTTGC	22	-6.02982241086787
CBG	6	10.9051126054603	ROBINS	27	32.3468620073079
CENTEL	26	-18.0658005791033	SAWAD	13	-1.12027044018635
CK	27	14.1071477738489	SCB	24	-10.3620951894532
CPALL	26	10.7553900736925	SCC	23	-4.46266371436592
CPF	24	-11.7864293399391	TASCO	28	81.1788459410811
CPN	34	11.9894393951306	TCAP	20	11.8232058304104
DELTA	41	-15.6154251639605	TMB	22	8.89598855310597
DTAC	28	-73.5184309361466	TOP	33	-9.81867133223752
EGCO	32	9.68047027598773	TPIPL	23	33.4659562766239
GLOW	40	-22.7263477470511	TRUE	29	-14.1852151728102
GPSC	0	0	TTW	26	7.53898056740626
HMPRO	25	29.9193976720908	TU	30	-0.347107252597664
INTUCH	23	12.1146894821578	WHA	24	19.5755362623513
IRPC	27	17.6259305858812	BEM	-	-

Table 4.5 Buy-sell simulation result from SMA of tick change.

Stock	Transactions	Profit (%)	Stock	Transactions	Profit (%)
ADVANC	25	-6.89380256763199	IVL	27	20.0254298535954
AOT	29	33.1786584104143	KBANK	28	1.86835601118035
BA	5	-1.12420495014176	KCE	21	69.1303538765374
BANPU	26	7.27944108662104	KTB	27	-5.9689192037344
BBL	36	-7.32377017385525	LH	24	-13.5314932979729
BCP	31	-9.67478688109438	MINT	38	16.3732379174924
BDMS	29	24.8640234172919	MTLS	7	-10.1281626503367
BEC	32	-56.3608189159935	PS	27	16.6596830258591
BH	25	57.7517373373587	PTT	27	20.1468832943797
BLA	30	-16.5397148247004	PTTEP	31	3.68695837841823
BTS	19	10.181644793919	PTTGC	22	-6.02982241086787
CBG	6	10.9051126054603	ROBINS	27	32.3468620073079
CENTEL	26	-18.0658005791033	SAWAD	13	-1.12027044018635
CK	27	14.1071477738489	SCB	24	-10.3620951894532
CPALL	26	10.7553900736925	SCC	23	-4.46266371436592
CPF	24	-11.7864293399391	TASCO	28	81.1788459410811
CPN	34	11.9894393951306	TCAP	20	11.8232058304104
DELTA	41	-15.6154251639605	TMB	22	8.89598855310597
DTAC	28	-73.5184309361466	TOP	33	-9.81867133223752
EGCO	32	9.68047027598773	TPIPL	23	33.4659562766239
GLOW	40	-22.7263477470511	TRUE	29	-14.1852151728102
GPSC	0	0	TTW	26	7.53898056740626
HMPRO	25	29.9193976720908	TU	30	-0.347107252597664
INTUCH	23	12.1146894821578	WHA	24	19.5755362623513
IRPC	27	17.6259305858812	BEM	-	-

Table 4.6 Buy-sell simulation result from EMA of closing price.

Stock	Transactions	Profit (%)	Stock	Transactions	Profit (%)
ADVANC	39	-20.3823983306502	IVL	43	-3.46354257427299
AOT	47	13.7335089847459	KBANK	33	14.4683980421824
BA	7	-1.13270627506854	KCE	49	44.7929742512673
BANPU	35	-14.0673302401327	KTB	46	-25.7836994108413
BBL	44	-17.6623435868428	LH	45	-20.5824392028888
BCP	49	-45.0544807547095	MINT	44	14.8242940704656
BDMS	39	30.906692695389	MTLS	13	-14.6157505108715
BEC	43	-73.5922115425042	PS	40	0.345330531662167
BH	39	42.3167044103691	PTT	38	27.132686124804
BLA	38	11.0173490322048	PTTEP	49	-26.6523347323412
BTS	43	1.06815699650958	PTTGC	46	-28.7709071808239
CBG	12	35.9546796741689	ROBINS	44	6.52665948677473
CENTEL	42	-25.4180415845941	SAWAD	16	-4.84723737055447
CK	42	1.35195004028906	SCB	33	-5.22928080013443
CPALL	52	-16.8421523007547	SCC	42	-15.638132905409
CPF	33	-3.85779907078727	TASCO	41	76.6728568438316
CPN	55	-12.0626636616499	TCAP	44	-7.95455518669151
DELTA	67	-56.2409090429292	TMB	41	-24.9049006998956
DTAC	37	-66.6730565442374	TOP	44	-18.130272762276
EGCO	53	-1.55894433276298	TPIPL	34	50.2510905938192
GLOW	52	-51.6828290747547	TRUE	36	23.419288842852
GPSC	1	-2.04081632653061	TTW	61	-32.0765114480112
HMPRO	44	2.32134890979896	TU	53	-24.688988845454
INTUCH	41	-8.19554207955696	WHA	42	15.2403280887208
IRPC	50	-8.61259984636508	BEM	-	-

Table 4.7 Buy-sell simulation result from EMA of tick change.

Stock	Transactions	Profit (%)	Stock	Transactions	Profit (%)
ADVANC	39	-18.1243338145212	IVL	46	-1.34385071460539
AOT	49	12.6925033539555	KBANK	34	11.1616572658029
BA	8	-1.96526991301322	KCE	49	44.7929742512673
BANPU	35	-14.0673302401327	KTB	46	-25.7836994108413
BBL	45	-19.4695320124523	LH	47	-32.7923887574249
BCP	49	-45.0544807547095	MINT	44	14.8242940704656
BDMS	39	30.906692695389	MTLS	13	-14.6157505108715
BEC	43	-77.3209115487094	PS	41	6.76672071642137
BH	39	42.3167044103691	PTT	38	27.132686124804
BLA	38	11.0173490322048	PTTEP	49	-26.6523347323412
BTS	42	0.0514904170523511	PTTGC	46	-28.7709071808239
CBG	12	35.9546796741689	ROBINS	44	6.52665948677473
CENTEL	42	-25.4180415845941	SAWAD	16	-4.84723737055447
CK	40	5.87745679746688	SCB	33	-5.22928080013443
CPALL	52	-16.8421523007547	SCC	42	-15.638132905409
CPF	33	-2.84565332179941	TASCO	41	76.3423820285132
CPN	55	-12.0626636616499	TCAP	44	-7.95455518669151
DELTA	67	-56.2409090429292	TMB	41	-24.9049006998956
DTAC	37	-67.9517796203696	TOP	44	-18.130272762276
EGCO	53	-1.55894433276298	TPIPL	34	44.736057681526
GLOW	52	-51.6828290747547	TRUE	36	20.9001281451814
GPSC	1	-2.04081632653061	TTW	60	-30.2073525695065
HMPRO	44	2.32134890979896	TU	53	-24.688988845454
INTUCH	41	-8.19554207955696	WHA	42	15.2403280887208
IRPC	49	-6.79416250112865	BEM	-	-

Table 4.8 shows the average profit from the above tables.

Table 4.8 Average profit of each indicator.

Indicator	Average Profit (%)
MACD	-4.35592563458567
SMA$_C hange$	5.66296750941681
SMA$_T ick$	5.66296750941681
EMA$_C hange$	-5.43002205317228
EMA$_T ick$	-5.70679370259828

Clearly from the results, the signal from SMA of closing change and tick change provided the best profit over our dataset. The results of SMA from closing change and tick change are the same because of how we calculate SMA and tick change.

$$\begin{aligned}
\text{For closing change} : SMA_m &= \frac{\sum_{i=0}^{19} change_{m-i}}{20} \\
&= \frac{\sum_{i=0}^{19} (close_{m-i} - close_{m-i-1})}{20} \\
&= \frac{close_m - close_{m-20}}{20} \\
\text{For tick change} : SMA_m &= \frac{\sum_{i=0}^{19} tickChange_{m-i}}{20} \\
&= \frac{\sum_{i=0}^{19} (tickCount_{m-i} - tickCount_{m-i-1})}{20} \\
&= \frac{tickCount_m - tickCount_{m-20}}{20}
\end{aligned}$$

Since tickCount is calculated from closing price:

$$\begin{aligned}
SMA_{change,m} \geq 0 &\Leftrightarrow close_m \geq close_{m-20} \\
&\Leftrightarrow tickCount_m \geq tickCount_{m-20} \\
&\Leftrightarrow SMA_{tickChange,m} \geq 0
\end{aligned}$$

So the signal from SMA of closing change is the same as the signal from

SMA of tick change.

In the next chapter, we present our proposed prediction models for forecasting stock price and stock price movement along with the experimental results on the dataset presented in this chapter.

CHAPTER V

THE PREDICTION MODELS

In this chapter, we are going to discuss the prediction models we constructed, and give the experiment results. All the results listed in this chapter are only from one combination of SVR and kernel function, a full list of results from all combinations can be found in the Appendix.

5.1 Price Prediction Model

First, we build the forecasting model to predict the stock price directly. All the results shown in this section are from EPS-BSVR with Bessel kernel, since it gives the best result for price prediction. The full list of results from all combinations of SVR type and kernel function can be found in the Appendix.

5.1.1 Time Series based Model

We start with a simple model that is similar to a time series model

$$x_n = f(n).$$

Combining with closing price, closing change, and tick change, the model becomes:

$$close_n = f_{close}(n)$$

$$change_n = f_{change}(n)$$

$$tick_n = f_{tick}(n)$$

Table 5.1 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

Table 5.1 MAPE results for time series based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	30.4365471873679	1.7588229751705602	1.73139816510763	IVL	16.192370807095898	2.10605124942168	2.17626566380856
AOT	31.1042932784334	1.0879963226789902	1.10583443366999	KBANK	30.3194047007787	1.42289569633995	1.36837150801671
BA	7.6352899847953095	0.917638088484441	0.9174214150854111	KCE	25.529177302622003	1.4633904895637702	1.46308997023758
BANPU	13.4689182750837	2.02133538335986	2.0219031656916098	KTB	34.315957776803	1.17281267572147	1.17281267572147
BBL	14.2859845338635	1.22651055409095	1.22633679193311	LH	21.9242495292987	1.26612418247049	1.2648261288087599
BCP	9.67757435520536	1.39297014109885	1.39297014109885	MINT	40.1056818904857	1.6138719336023999	1.62098206267357
BDMS	24.0034522360619	0.8835290149874809	0.883529014987482	MTLS	3.19258603243593	0.8887300848413431	0.888730084841341
BEC	5.65810745218769	1.6852916159861902	1.6789430143867201	PS	47.354683311886	1.28182856157774	1.4015608768278298
BH	26.7758639786985	1.39285955811169	1.40481477357646	PTT	24.2382467833181	1.84458886512791	1.8433324084315998
BLA	14.4698628456331	1.6158618488910401	1.6158618488910401	PTTEP	30.245399203498803	2.23636379757232	2.2520782433001902
BTS	5.99539637057194	0.938210403121603	0.9344968876878971	PTTGC	13.4505016112789	1.7952095455600998	1.7952095455600998
CBG	23.3572365782905	2.0783091383135996	2.07830913831361	ROBINS	29.6427084780426	1.47207449492774	1.47207449492774
CENTEL	9.67385787139821	1.53665431291177	1.53767407844862	SAWAD	15.536675695657202	1.3998203810663	1.3998203810663
CK	17.3613563391688	1.37587062555614	1.4097855280832299	SCB	25.7220434998062	1.60530632943926	1.60530632943927
CPALL	16.1208446677134	1.356500186372	1.356500186372	SCC	9.05190492015798	1.14509971780233	1.14583775021249
CPF	19.959756523421802	1.9858304063036898	1.98743711458674	TASCO	65.9610389652857	2.72137524510087	2.85453728735258
CPN	11.6725359670682	1.196877561173 61	1.19687756117361	TCAP	19.5837320474051	1.1785035912131399	1.1785035912131399
DELTA	17.7008852716586	1.79976991294817	1.79976991294817	TMB	31.5290788086688	1.39117229532014	1.39131190339141
DTAC	27.103600104781002	3.1374578174826904	3.11023392692726	TOP	11.8563353074461	1.50544365299571	1.50544365299572
EGCO	19.3967903888865	0.94574401842587	0.9527705824848159	TPIPL	13.899360121834	1.62018088616484	1.6471725109811102
GLOW	9.49338890053736	1.30005838936994	1.2999096774010699	TRUE	29.7416361986782	2.0445334841503002	1.9833989517123298
GPSC	4.76251631658221	1.6439541054089302	1.64395410540892	TTW	8.61193793413116	0.7985957531673961	0.7949726109553759
HMPRO	38.3951430696941	1.47167928809178	1.46773771386986	TU	21.3025475793225	1.5942322389226	1.5942322389226
INTUCH	30.243864961346702	1.5845948328745099	1.5845948328745099	WHA	11.3443556949468	1.55285753711673	1.55335797819801
IRPC	9.68168998139058	1.33207773479281	1.39265265909979	-	-	-	-

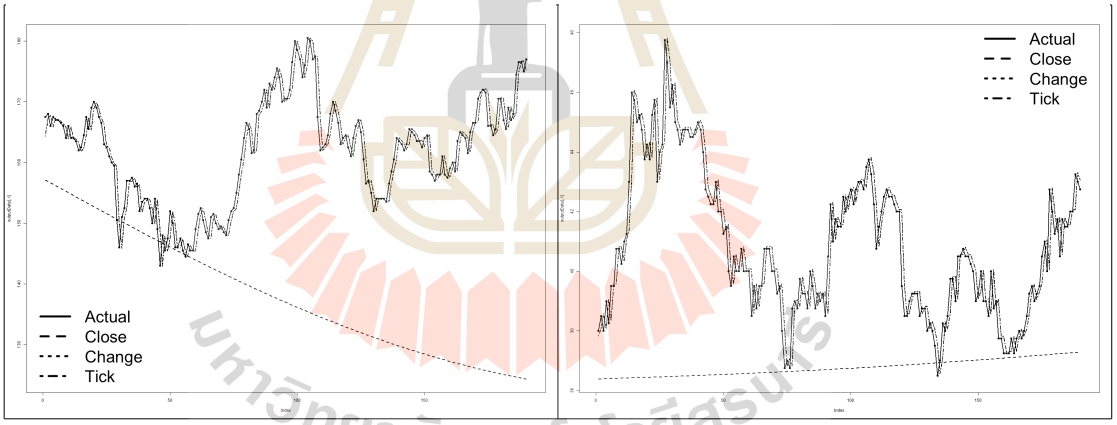

Figure 5.1 Graph of predicted results from two stocks for time series based model.

Table 5.2 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. It is clear from the result that the model from closing change and tick change gives better result than the model from closing price. The closing change model gives slightly better results than the tick change model.

Table 5.2 Average error for time series based model.

Base	MAPE	MSE	R^2
Close	20.7976810538923	962.342226878783	-11.7629184314798
Change	1.526274835208	4.30321442368275	0.928369867305664
Tick	1.5327539901981	4.32411188609181	0.927716804994936

5.1.2 Historical data based Model

Next, we construct the model by using the historical data

$$x_n = f(x_{n-1}, x_{n-2}, ..., x_{n-i})$$

We use a week of historical data, e.g.,the past 5 days, and do prediction for the day ahead. So the equation becomes:

$$x_n = f(x_{n-1}, x_{n-2}, ..., x_{n-5})$$

Combining with closing price, closing change, and tick change, the model becomes:

$$close_n = f_{close}(close_{n-1}, close_{n-2}, close_{n-3}, close_{n-4}, close_{n-5})$$
$$change_n = f_{change}(change_{n-1}, change_{n-2}, change_{n-3}, change_{n-4}, change_{n-5})$$
$$tick_n = f_{tick}(tick_{n-1}, tick_{n-2}, tick_{n-3}, tick_{n-4}, tick_{n-5})$$

Table 5.3 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

Table 5.4 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. Similar to time series model, closing change and tick change give better results than closing price model. While the closing price model gives better error than the time series model; closing change and tick change give higher error than time series model.

Table 5.3 MAPE results for historical data based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	34.1661930645251	1.8337263226365599	1.7653989143993802	IVL	8.66285711195782	2.160598062161 56	2.20847037675511
AOT	22.6526246848853	1.07327581531368	1.07421105875719	KBANK	2.95145542014278	1.5114178635946198	1.51030672755015
BA	1.3481813620679401	0.9242147293998569	0.8987312268067951	KCE	31.4699476950623	1.50977344037323	1.51804894209248
BANPU	33.0861091160169	2.05347314676642	2.04321745563109	KTB	1.56410173934475	1.23275838821613	1.23266538648107
BBL	4.91962763569944	1.24873247327697	1.25987581180874	LH	1.71896746616768	1.25204341025505	1.26442743209616
BCP	1.8756179631395202	1.39229481566405	1.39229481566405	MINT	14.024911493999001	1.70016289868579	1.68306394100547
BDMS	9.2322559748852	0.900515105112998	0.9005053817005251	MTLS	3.3634092330281797	0.8473301693651221	0.8475575139447069
BEC	23.9683178031492	1.77424740176886	1.7684189240752601	PS	1.6991232944841599	1.27771673415334	1.26282019925851
BH	2.0893783698154302	1.50319164389001	1.50274415807681	PTT	4.88037216160889	1.86524921482779	1.86588448142695
BLA	12.0927812820157	1.7420269671692301	1.74204211584119	PTTEP	11.8620078147084	2.1395940074090603	2.19575846517247
BTS	1.24242509353287	0.981851570194498	0.9788221929719809	PTTGC	3.7924270055862297	1.86961859494652	1.86961859494652
CBG	27.4451115467016	2.07888366934899	2.07888366934899	ROBINS	4.69461307348781	1.56713230714488	1.56713230714488
CENTEL	5.717880976645881	1.60893414861323	1.61551184982356	SAWAD	2.1114417257657703	1.44766583065983	1.44768716888394
CK	2.73217721388711	1.4324377960913501	1.45508515402217	SCB	6.88462409420854	1.49200059418767	1.49200059418767
CPALL	2.3458141236036503	1.38634366352402	1.38634366352402	SCC	1.47153046194564	1.13121462051159	1.1295202311914099
CPF	3.2154486518338796	2.06277853208702	2.0259681554242697	TASCO	3.6920190215654	2.59702729987574	2.42533998497378
CPN	10.3098841853849	1.21300964641967	1.21300964641967	TCAP	2.2283645656983397	1.18088516158916	1.18088516158916
DELTA	2.22310040313923	1.78456648020438	1.78456648020438	TMB	1.73150320065474	1.43765719667613	1.44046212212633
DTAC	101.74892467985399	3.1400512272666603	3.1499122510963	TOP	17.0422004377337	1.52460409330827	1.52460409330827
EGCO	5.81103090346083	0.948105948008151	0.9475917969984889	TPIPL	2.32005772510501	1.69088931948515	1.72211104447598
GLOW	1.68365585709358	1.38128292006791	1.37599070523105	TRUE	3.0384149314463897	2.02310179186619	2.01062607322131
GPSC	5.51324148486756	1.69088672173253	1.69088672173253	TTW	1.09270546123772	0.828029985252308	0.832190753338897
HMPRO	4.54281592083961	1.51056367644184	1.51316409719114	TU	1.6991068161040301	1.38283932286505	1.38276313495662
INTUCH	29.800863087115896	1.67695732715575	1.67695732715575	WHA	1.98009369794412	1.5936224967113	1.5857443253431798
IRPC	6.50609914317916	1.31475607462628	1.3103644864309	-	-	-	-

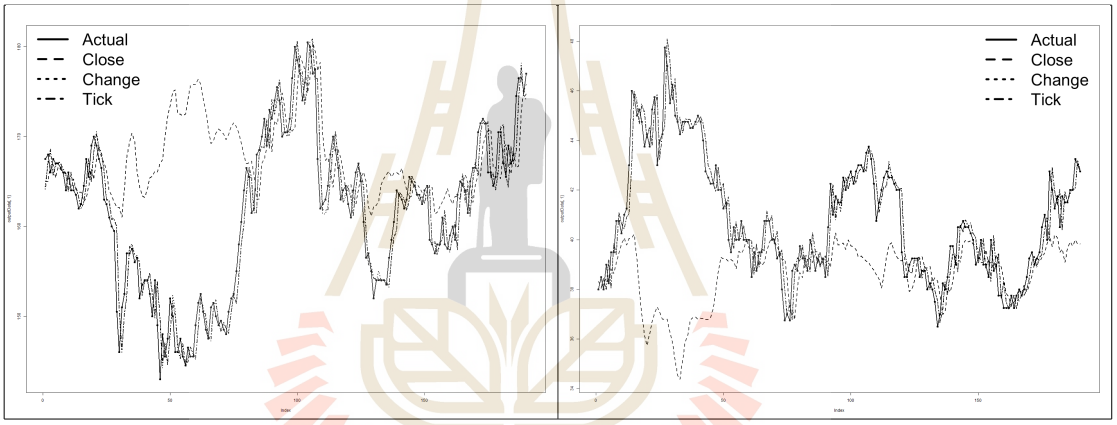

Figure 5.2 Graph of predicted results from two stocks for historical data based model.

Next, we will include data from technical analysis into the model.

5.1.3 Technical Analysis based Model

There are many indicators in technical analysis. We only use indicators that are commonly used in the market:

(1) EMA from 10 days (2 weeks), 60 days (3 months), and 250 days (1 year)

(2) Volume

(3) Buy Volume, Sell Volume

Table 5.4 Average error for histrocial data based model.

Base	MAPE	MSE	R^2
Close	10.0458329831903	389.220688885388	-1.48585555871906
Change	1.5493885842225	4.36002451449695	0.927283719981271
Tick	1.5459221860369	4.37100076248942	0.927485279956935

With EMA, the equations become:

$$close_n = f_{close}(close_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

$$change_n = f_{change}(change_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

Table 5.5 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

Table 5.5 MAPE results for EMA based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	32.97599447549 71	1.74586971915374	1.7316737297676899	IVL	7.52041863448252	2.0909694690030003	2.0711246405760697
AOT	22.515530004062	1.09768475342503	1.1084016830827599	KBANK	2.16875470222 50198	1.40620562624141	1.40807275961519
BA	1.56005198097914	1.16463936040711	1.04625705894804	KCE	30.9540425512714	1.50988469573527	1.50805838884955
BANPU	50.9606578956985	2.12702790205739	2.11234154512973	KTB	4.02819762598877	1.1932899538239399	1.19326312146879
BBL	8.459044672508801	1.27065166326475	1.26608337353365	LH	1.84030628989479	1.2569138293050799	1.23560126000666
BCP	2.13569551547237	1.41655346434848	1.41663661478677	MINT	13.359623796527401	1.80417087989913	1.8128574044394898
BDMS	10.1411666120352	0.893873717025034	0.894002149980252	MTLS	3.23190595123364	0.9506025252258221	0.950577438963093
BEC	28.6464989889308	1.8358725373790699	1.7620296436162	PS	1.34625415524365	1.28788053299793	1.25531185629616
BH	1.9407685107961599	1.48971465542334	1.47389573627174	PTT	10.6459472577278	1.8232659580262298	1.82515065141583
BLA	9.248960509135	1.7866053939301603	1.78645069489482	PTTEP	20.9522735203729	2.24061731124146	2.25416877005029
BTS	1.54221424352132	0.9521749059428869	0.9430722966254791	PTTGC	4.162156527496 46	1.8021412988927301	1.8021412988927301
CBG	27.2352079203917	1.9924281425678598	1.9924281425678598	ROBINS	6.09317207638209	1.6568364961740498	1.6568364961740498
CENTEL	12.545404016042799	1.7713101572396601	1.7427851167167299	SAWAD	2.00863772883699	1.502080730261	1.50212252125452
CK	2.3927851589019	1.37268545064241	1.38795423001349	SCB	17.2540874446363	1.4781112650860901	1.4781112650860901
CPALL	2.21776123405468	1.36383016744185	1.3638273152807099	SCC	1.2689163957168201	1.08934951902626	1.0883033692686002
CPF	6.4381299224402095	2.08466742125798	2.1655610008806	TASCO	15.6273699018892	2.92914151100671	3.4627643893750304
CPN	10.1442348561362	1.251979403146 24	1.251979403146 24	TCAP	3.07433012349232	1.16626152809041	1.16626152809041
DELTA	1.8618432843828199	1.8534251011693401	1.8534251011693499	TMB	1.6427131086145599	1.42981191518495	1.4292396840366
DTAC	136.941738125046	3.313773773833 08	3.2125761854548	TOP	17.5875868496184	1.5674001771400 3	1.5674001771400399
EGCO	6.33195962572631	0.984441825570044	0.982916057377875	TPIPL	2.15184177847484	1.76757029935674	1.74560117510175
GLOW	1.8460959565161499	1.36966684000729	1.38140215397409	TRUE	4.32110406957895	2.01037997088082	1.99303139992946
GPSC	3.6277231678820097	1.55507878111947	1.55507878111947	TTW	0.9012269371013361	0.811405580847697	0.813503123625307
HMPRO	4.73392621982002	1.51769347061087	1.52129335739763	TU	1.69652972388274	1.3728022580437	1.37253770151654
INTUCH	29.207963921275898	1.6669250668125999	1.6669250668125999	WHA	3.3912339026666896	1.5448254625485	1.53306403153553
IRPC	12.008802559352901	1.34778604035266	1.38571327550963	-	-	-	-

Table 5.6 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. This model gives higher error than the previous models.

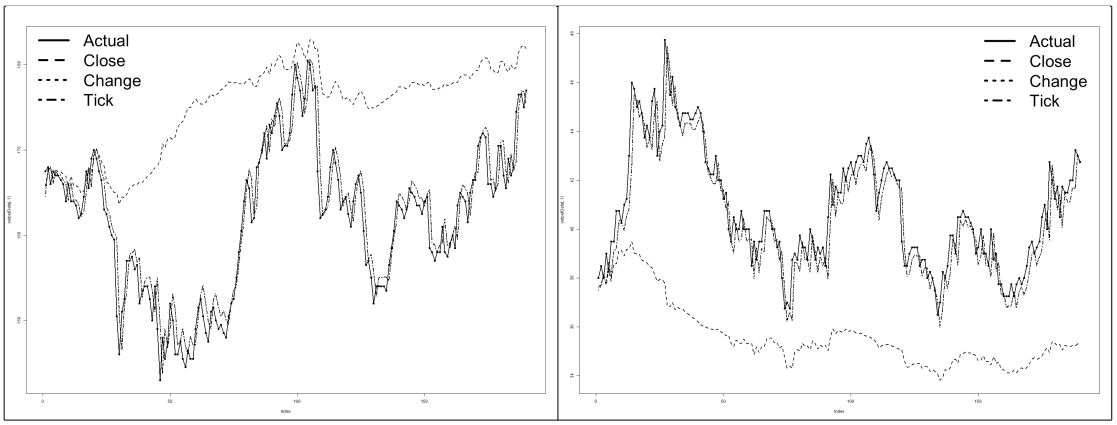

Figure 5.3 Graph of predicted results from two stocks for EMA based model.

Table 5.6 Average error for EMA based model.

Base	MAPE	MSE	R^2
Close	12.3446691924641	436.522731964164	-2.29534661035772
Change	1.569760785881	4.32731003230386	0.925161192808782
Tick	1.5740778401381	4.3499954704043	0.926057481640903

With Volume, the equations become:

$$close_n = f_{close}(close_{n-1}, vol_{n-1})$$

$$change_n = f_{change}(change_{n-1}, vol_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, vol_{n-1})$$

Table 5.7 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

Table 5.8 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. This models gives the best results so far for all based data. Tick change does slightly better than the closing change model.

Next, instead of using a single combined volume data, Buying Volume and

Table 5.7 MAPE results for volume based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	28.9273118474747	1.72716255566943	1.7445550291914198	IVL	5.60567741107312	2.10087072142413	2.08620333235242
AOT	15.0393892666282	1.04357535415925	1.03792074342421	KBANK	2.3563250423948903	1.3789722916426	1.3762084511542398
BA	1.05520905741966	0.9395112726488729	0.9222192216938621	KCE	23.6716714140037	1.4567262110319	1.45521815453941
BANPU	15.824131747303898	2.00156582460093	2.0008590617705604	KTB	1.30571449148452	1.1546367780173201	1.1546367780173201
BBL	3.2196366682712503	1.23182957147908	1.23000572139383	LH	1.34587873001486	1.21347639009009	1.22132949443704
BCP	1.4174332371454201	1.3477371949713801	1.34774609903657	MINT	9.02415760354629	1.5708644100492102	1.58316190805721
BDMS	6.121952875046	0.883687175008966	0.8836871750089671	MTLS	2.43454202663743	0.933724230190396	0.9337242301904041
BEC	15.7211683274499	1.68002053711271	1.6745726323571901	PS	1.36405062062809	1.24271619511683	1.24599505772675
BH	2.15365451068905	1.4106585432099699	1.41713398981337	PTT	4.00361231214187	1.8308749126586799	1.8323009593662998
BLA	7.602945141495191	1.64872690363824	1.64872690363977	PTTEP	8.15002969934697	2.3019116733834	2.23869466145797
BTS	1.04770196118152	0.918829221489997	0.912170512506446	PTTGC	2.68975973862763	1.77453964561276	1.77453964561276
CBG	27.135681986530702	1.99377400181171	1.99377400181171	ROBINS	3.04546777267317	1.46167125188103	1.46167125188103
CENTEL	3.4711166868772203	1.48010870039346	1.47982377020097	SAWAD	1.49704837521322	1.3922467606174	1.39228412835123
CK	2.30608895230823	1.34091814119425	1.35361529437848	SCB	4.85154441249796	1.47501707923407	1.47501707923407
CPALL	1.7405593908976997	1.3262250083793998	1.3262250083793998	SCC	1.12529228818569	1.08013454325351	1.07896775992242
CPF	3.32402737355585	2.01536349495965	2.0177705624485	TASCO	4.32886658451767	2.34290404001167	2.32916155447904
CPN	7.471519479781399	1.22831309044698	1.2283130904469899	TCAP	1.9666846054674498	1.1305878542863799	1.13055953027442
DELTA	1.9225686579395	1.7121915215183199	1.7121915215183199	TMB	1.46260372724759	1.38707280638982	1.39066418423034
DTAC	57.911805388245696	3.07875657244066	3.07098278813259	TOP	13.3746499207061	1.49360744178871	1.49360744178871
EGCO	4.15812451695314	0.913333078435806	0.913580760208917	TPIPL	2.17836168228789	1.59044683583535	1.6261858368641802
GLOW	1.3011300275173598	1.25507891260768	1.25549118754809	TRUE	2.73178163924138	1.97333209122949	1.97652955276277
GPSC	1.97185572950573	1.48704640946769	1.4870464094676799	TTW	0.9865166331427949	0.8113521597321901	0.808893365343977
HMPRO	3.52977810414835	1.46336421080879	1.4628851766087299	TU	1.39413048252005	1.3281030274874501	1.3281030274855001
INTUCH	24.932129711737 4	1.6038282107231598	1.6038282107231598	WHA	1.73264251987046	1.51447106460471	1.51149857952694
IRPC	5.25307663717321	1.28099258308228	1.28269184830868	-	-	-	-

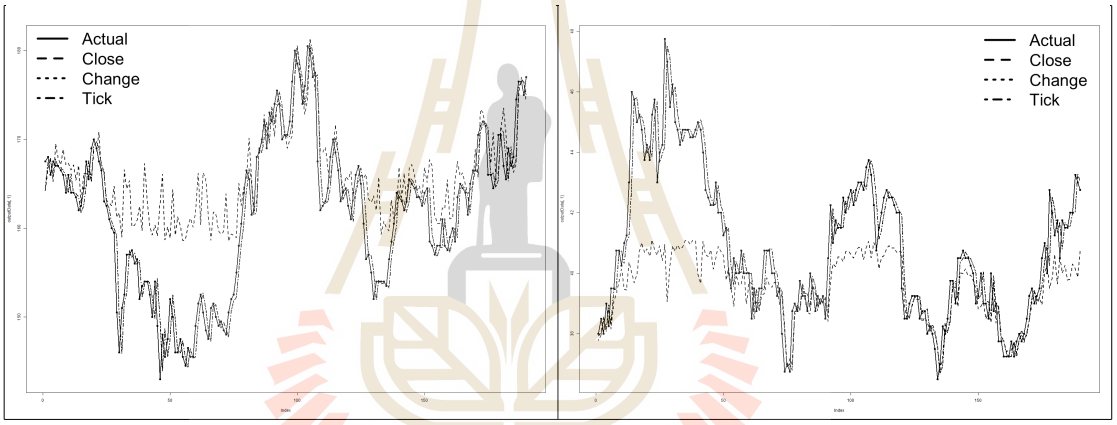

Figure 5.4 Graph of predicted results from two stocks for volume based model.

Selling Volume are used. So the equations become:

$$close_n = f_{close}(close_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$change_n = f_{change}(change_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

Table 5.9 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

Table 5.10 shows the average error of EPS-BSVR with Bessel kernel from

Table 5.8 Average error for volume based model.

Base	MAPE	MSE	R^2
Close	7.0854491227908	199.341061676753	-0.26036291835651
Change	1.4888339490985	4.16812754841777	0.93123276226303
Tick	1.4880198507158	4.15571623426123	0.931277232667798

Table 5.9 MAPE results for buy-sell volume based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	31.656504562224402	1.74598775969472	1.74554576888146	IVL	6.338783370815521	2.11419196552121	2.0822782188432196
AOT	17.9746799256719	1.0397323449821498	1.04955211848392	KBANK	2.6493574585144697	1.38585609556605	1.38904896847295
BA	1.00980998586177	0.9233789051474169	0.8908831853864639	KCE	31.626551374023197	1.45692082887405	1.45310626084876
BANPU	21.7965984060539	1.9910196544524899	1.9922986451479803	KTB	1.7589954596944 2	1.16076024118612	1.16083009157955
BBL	4.13207900644359	1.2177536308838	1.2149941835031601	LH	1.53274555294026	1.2308075416433	1.23590949746158
BCP	1.47190486756588	1.3515756333524098	1.3515756333524098	MINT	11.0243159668997	1.61710532931027	1.62063307789839
BDMS	7.79622522792512	0.910788346806474	0.9107800424623381	MTLS	2.59866552688668	0.971951114280661	0.971954088193359
BEC	21.1855152800326	1.6762890490579399	1.6722734825224599	PS	1.4553490900289698	1.2702467905538999	1.29498073898714
BH	2.32967956673354	1.41451502579046	1.42071977282614	PTT	4.62689061152903	1.8632868096166402	1.8659562028901198
BLA	9.30436481645015	1.6200231431272498	1.62002002005769	PTTEP	12.383791054252399	2.32668824391282	2.26885891960446
BTS	1.16538845221584	0.930145610368072	0.930265963122604	PTTGC	3.52965870870197	1.78961087958252	1.78961087958252
CBG	27.4500769187018	1.9831869225907899	1.9831869225907899	ROBINS	4.03102107252127	1.46885901556117	1.46885901556117
CENTEL	4.81559590068004	1.51581826824538	1.51425141152649	SAWAD	1.38465924964199	1.42301491737861	1.42301410554674
CK	2.73239840930365	1.35114300348953	1.3612663148056399	SCB	6.2282563529112895	1.44114099701832	1.44114099701832
CPALL	2.06081512914446	1.33517780647776	1.33517780647776	SCC	1.1931597500012	1.07509795514249	1.0802992918768402
CPF	3.87652020136983	1.9988050995009998	2.01690572954397	TASCO	5.93518362419908	2.34831200584323	2.37110934498561
CPN	8.21577573142942	1.23530429187387	1.23530429187387	TCAP	2.47279256807438	1.1409924323149	1.1409924323149
DELTA	1.99457282534189	1.74838851644462	1.74838851644462	TMB	1.6533830718328102	1.4159355050806	1.416390721406
DTAC	77.1995157464266	3.10021586360058	3.1062189987729303	TOP	15.751149899192802	1.49219675541249	1.49219675541249
EGCO	4.77916326441486	0.9183014594168429	0.918416194129869	TPIPL	2.5239940322253798	1.6199185091345498	1.6281764760167
GLOW	1.45744511602032	1.27104302810668	1.27643843010792	TRUE	3.3658512875497695	2.0419490935874	2.01529388247909
GPSC	2.26927884212806	1.62340784780758	1.62340784780756	TTW	1.16124152492528	0.80869031882523	0.810686648820479
HMPRO	4.23410698466182	1.4960921723338299	1.4948834877282702	TU	1.42932829121767	1.3239715739653	1.32399246085608
INTUCH	28.593887310910898	1.61976015935011	1.61976015935011	WHA	1.7659994350103398	1.56620777002469	1.5640596965479698
IRPC	6.38024683089729	1.28791252234752	1.28368014224635	-	-	-	-

all stocks by based data. While this model gives a pretty good result, it gives slightly higher error than using volume data alone. This shows that using more data may not help improving the forecasting performance.

Table 5.10 Average error for buy-sell volume based model.

Base	MAPE	MSE	R^2
Close	8.6592504824939	272.487321085218	-0.790024415613916
Change	1.5032546684609	4.18710235880524	0.929250108480984
Tick	1.5025627314767	4.19502193270693	0.929304519007407

In addition to these three models, we also construct forecasting models from the other technical analysis data. The remaining technical analysis based models and their experimental results can be found in the Appendix.

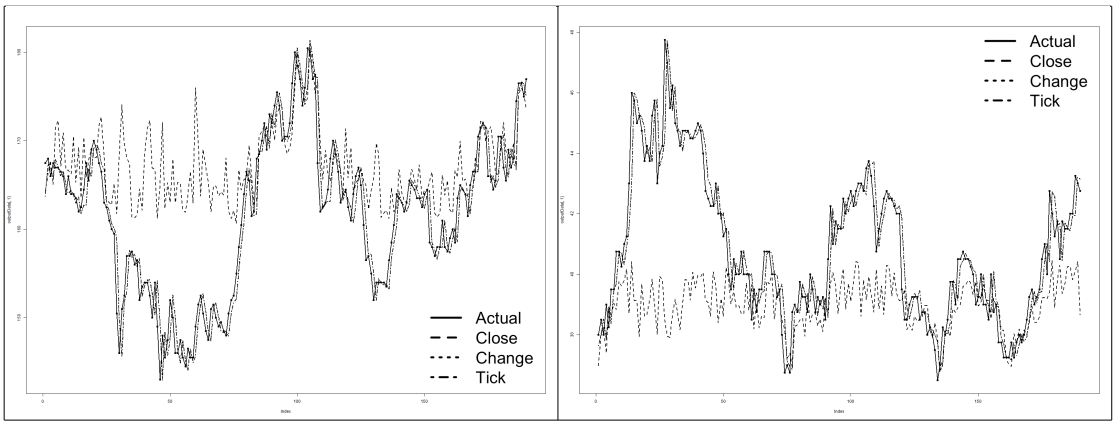

Figure 5.5 Graph of predicted results from two stocks for buy-sell volume based model.

5.1.4 EMD based Model

Next, we are going to employ EMD to preprocess the data, and use the resulting IMFs for forecasting. First, we used IMFs of closing price.

$$close_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.11 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

Table 5.12 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. It gives a higher error results than the previous models for all based data.

Table 5.11 MAPE results for EMD of closing price based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	47.6309700275779	3.19473116010677	2.3887873894902603	IVL	17.6928975718319	2.68521782235742	2.78536674872244
AOT	37.8251721399479	1.19177969230693	1.28636037745151	KBANK	7.74720591666079	2.1929111710244897	1.8151214378646
BA	1.60231446312321	1.64050537469103	1.41772342395394	KCE	53.767206714432405	1.6260519169129701	1.65612745025775
BANPU	80.03981955413501	4.54097686977677	4.48034738664169	KTB	12.6805239605841	1.40643773893699	1.4061128924913502
BBL	14.9335980605518	1.373474893335	1.3610468593923901	LH	4.80743338125847	1.9962780916645801	1.7661972087080198
BCP	5.81218870570734	1.71908343726555	1.71908343726555	MINT	31.3658242498079	2.08212740293788	2.02866037135004
BDMS	20.0308683151632	1.15574732703904	1.1557873117038	MTLS	4.53801018081293	0.9939422552524211	0.993875516572008
BEC	65.5151089781146	3.08272676254993	2.56206444425629	PS	5.71080680251911	1.6755577563528599	1.5833599408406
BH	6.56873178973629	1.7150179003985502	2.21524622395133	PTT	25.1484435231825	2.2568110386837903	2.23704678627888
BLA	29.058951518846897	2.57807330112503	2.5777125438298	PTTEP	51.0564184586469	3.9196641571240196	2.78276533049292
BTS	10.3316179326499	1.70733543854267	1.58634283152082	PTTGC	17.5669937261731	2.84637119330078	2.84637119330078
CBG	27.7037284920314	2.46544933332403	2.46544933332403	ROBINS	18.7023245501588	1.81853437147614	1.81853437147613
CENTEL	16.295543467311898	2.39552864812695	2.38551905932613	SAWAD	5.74174633124746	2.0192323934420697	2.01933163116387
CK	12.2300332925389	1.9609917475603402	1.74722870794322	SCB	21.3167914502143	1.82880506771723	1.82880506771723
CPALL	8.42456510170328	1.5699409998576102	1.5699409998576102	SCC	4.57253746169297	1.3633452473224699	1.36861573573722
CPF	30.3072344161367	2.7861728579680602	2.51296362172079	TASCO	24.2805966771624	2.5118712423230503	3.18640400334454
CPN	15.1374682173129	1.61160311027588	1.61160311027588	TCAP	8.46694540177466	1.3400227268629301	1.34002272686292
DELTA	6.22186434784107	2.67530021554931	2.67530021554931	TMB	6.08847277167473	1.7823046370323898	1.78551228551009
DTAC	254.86325291588398	3.96570101358361	3.5572456910241397	TOP	26.101697680576603	3.30959885480063	3.30959885480063
EGCO	23.509490470394702	1.15714796638523	1.1693636442781798	TPIPL	10.1600128058687	2.17102530867397	2.4344546172175097
GLOW	3.7673415857936896	1.4812154717406	1.4817241171788	TRUE	12.6969139281677	2.7549766004045	2.34341526848955
GPSC	1.9264617219534301	1.6422334688541902	1.6422334688541902	TTW	5.2438920898604895	1.2055728408912498	1.30277145707012
HMPRO	13.6278566142849	1.6931587748108001	1.6753872367607898	TU	8.03637369225709	3.2467801672404497	3.24707481132868
INTUCH	34.5674474720345	1.8448818720817801	1.8448818720817801	WHA	4.34675701279628	2.21103724616251	2.20678411371712
IRPC	26.520728846401898	1.79341153802032	1.97538492636046	-	-	-	-

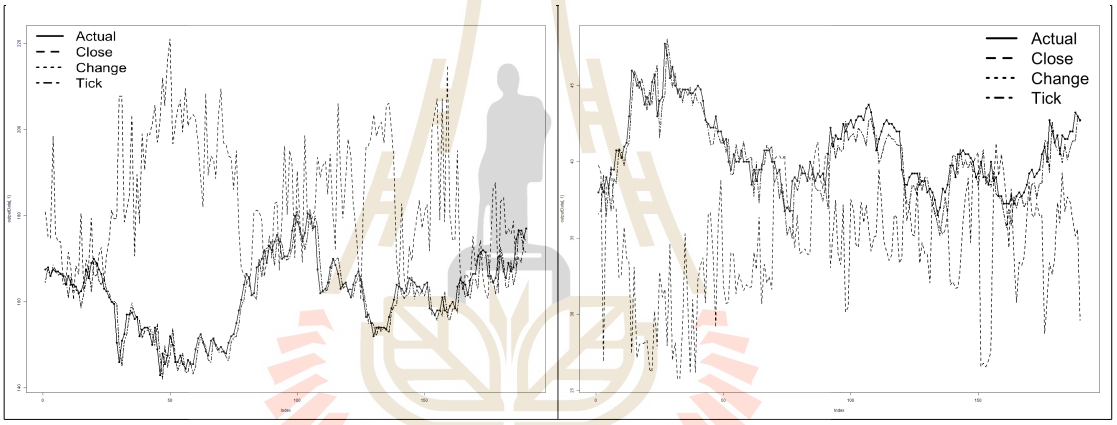

Figure 5.6 Graph of predicted results from two stocks for EMD of closing price based model.

Next, we construct the model from IMFs of closing change.

$$change_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.13 shows the MAPE results of this model with EPS-BSVR and Bessel

Table 5.12 Average error for EMD of closing price based model.

Base	MAPE	MSE	R^2
Close	24.1283507099294	1218.95230368568	-9.60829223500083
Change	2.1262584984525	6.58408563459305	0.872130284876349
Tick	2.0644297562308	6.44912022709155	0.87667956765862

kernel.

Table 5.13 MAPE results for EMD of closing change based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	48.6403367172731	2.66766808708634	2.21012584104569	IVL	21.5308958933621	3.5192080591016404	3.6507913728847097
AOT	26.3396667277918	1.28540866717568	1.46209073805555	KBANK	21.701438467618	1.62600799458418	1.5210810583649301
BA	1.7452808935252602	1.22710390950228	1.16418522275884	KCE	57.881549106784504	1.60639252449849	1.6484466223424001
BANPU	95.9350146571969	2.7916830256646	2.8310156371476403	KTB	12.1272729071109	1.65957252140344	1.65972400535829
BBL	19.6610052231535	1.37792625669852	1.37928130701652	LH	4.30404437476074	2.02420125076678	1.84263753313589
BCP	6.09894994540152	1.8635193996692299	1.8635193996692299	MINT	32.5040493442449	2.17113570356162	2.6759155417716203
BDMS	12.5242413115317	1.1631759558527	1.16312408771993	MTLS	3.6305026827792295	1.18924957203028	1.18929707900899
BEC	64.8493733683731	3.17640108624541	2.65236761669071	PS	17.9921089771717	1.9231576237189603	3.07043074317132
BH	12.523091838680301	1.5714486670707801	1.88438611360928	PTT	22.9051889834986	2.30673197710119	2.30063564837461
BLA	29.2207989154406	2.14135625657836	2.14127937892419	PTTEP	64.7270520912939	2.65236755009885	2.39164912703345
BTS	7.47345861094887	1.5598021742202	1.48537952166344	PTTGC	16.127354368726397	2.41550425219447	2.41550425219447
CBG	29.015297153262598	1.99170743160155	1.99170743160155	ROBINS	14.384158732225599	2.0829270526520003	2.0829270526520003
CENTEL	25.8919056348152	2.1434869958138902	2.17246310301234	SAWAD	2.9982940585999303	1.57270017381517	1.57263132112789
CK	17.408987332498	2.62245953547933	3.11745601209622	SCB	17.259755103316397	1.45356541787482	1.45356541787482
CPALL	8.3168715973838	1.48384200809295	1.48384200809295	SCC	19.9159577845686	2.30534004104026	2.3818566630395903
CPF	20.723987129038502	2.35312841403247	2.37485383150945	TASCO	35.9197892251316	2.56661176367252	3.1293927676955997
CPN	16.7398510056753	1.35832525889927	1.35832525889927	TCAP	8.83621648430655	1.25013976176691	1.25013976176691
DELTA	6.612089873376539	2.3876049585544	2.3876049585544	TMB	16.0723704063221	1.87458707945257	1.8668706673022
DTAC	230.77772623096303	4.473542747549089	4.39992725904806	TOP	22.0253013917557	1.7444108294053602	1.7444108294053702
EGCO	15.084079601675299	1.18956909524424	1.2013316929392301	TPIPL	39.089595727532	2.7379426036232504	3.141294526495558
GLOW	5.18501033236035	1.773151864187773	1.7680021481591999	TRUE	15.644774288751801	2.97016406600085	2.53325784904105
GPSC	2.3743190650515102	1.61000950177373	1.61000950177373	TTW	5.14307336431069	1.438195255308	1.5472365665337
HMPRO	18.1346237187139	1.77176575047738	1.7936812985358201	TU	8.28348512583347	1.8109873585789402	1.81095775904574
INTUCH	54.4554791334238	2.10045399948516	2.10045399948516	WHA	18.3667330504187	3.1475327476721597	3.2081760382726103
IRPC	18.7822988887516	1.4644328825119501	1.53787171674748	-	-	-	-

Table 5.14 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. It gives better results than using EMD from closing price for closing change and tick change model, but gives higher error for closing price model. It still gives higher error results than the previous models for all based data.

Table 5.14 Average error for EMD of closing change based model.

Base	MAPE	MSE	R^2
Close	26.4058103438108	1358.77863487358	-12.8998394908434
Change	2.0326042267223	8.8068114647496	0.878126391451077
Tick	2.0739411283398	6.01715986167739	0.856914478687777

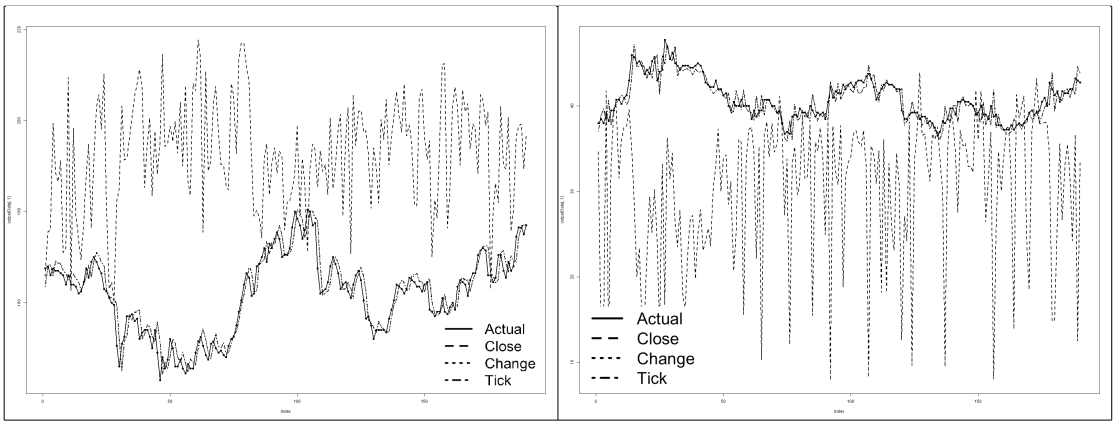

Figure 5.7 Graph of predicted results from two stocks for EMD of closing change based model.

Lastly, we use IMFs of tick change to build the forecast model.

$$tick_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.15 shows the MAPE results of this model with EPS-BSVR and Bessel kernel.

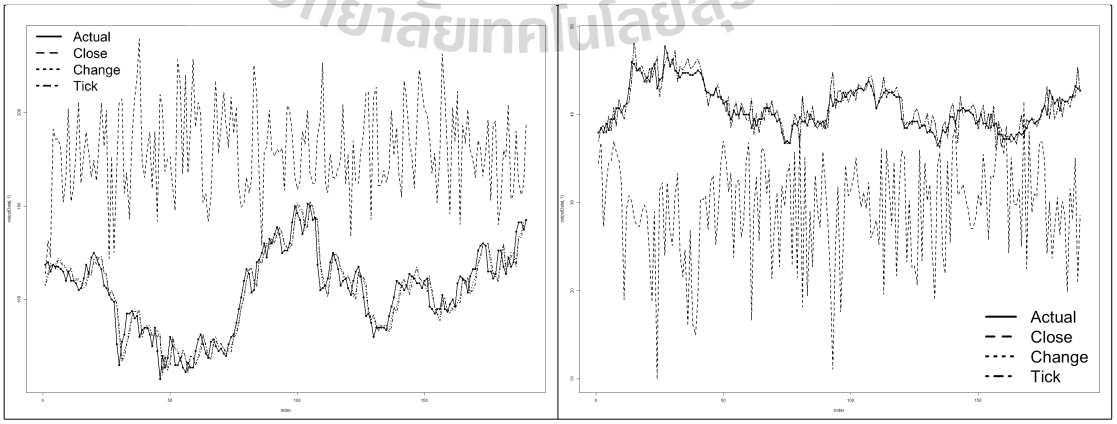

Figure 5.8 Graph of predicted results from two stocks for EMD of tick change based model.

Table 5.15 MAPE results for EMD of tick change based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	51.0050312182148	2.49542898035962	2.0043321744714198	IVL	24.7307154833486	3.5738665777075496	3.66188573348933
AOT	46.444118171536	1.27856702129367	1.47710905519768	KBANK	26.057300955259098	1.85372766010197	1.71597634488304
BA	1.55500191729788	1.10722019011378	1.0818879502299201	KCE	61.876677623370504	2.10994048857092	2.13281737242028
BANPU	96.1679517517802	2.76549528576444	2.82094778494291	KTB	8.733482769125791	1.5605435718529899	1.56033413732126
BBL	18.873226839418102	1.4382659761260101	1.54365041761946	LH	5.36196313882199	1.9280934276252701	1.75096065416223
BCP	6.09894994540152	1.8635193996692299	1.8635193996692299	MINT	35.4146116697643	1.93080959467922	1.9686416908424698
BDMS	11.6912878792158	1.33737324033816	1.33749919406857	MTLS	3.6281107557626404	1.19017798231146	1.1900973175501202
BEC	67.3932694734089	2.99557574239929	2.57691873028377	PS	36.5913742169084	2.1282324661368204	2.29314683552633
BH	17.1815159895901	1.52248614913542	1.56767579182941	PTT	23.1223449524637	2.30238242186631	2.29034537955682
BLA	22.7245193016114	2.31389686922528	2.31402869506545	PTTEP	87.2367705532665	2.6748525968710397	2.41908702046275
BTS	3.49587804544423	1.5819594397709	1.53960018589419	PTTGC	16.127354368726397	2.41550425219447	2.41550425219447
CBG	29.015297153262598	1.99170743160155	1.99170743160155	ROBINS	14.384158732225599	2.0829270526520003	2.0829270526520003
CENTEL	27.9793487656044	2.41638013150521	2.44560837160274	SAWAD	2.8987799067132203	1.5848404249156698	1.58456874052374
CK	17.6712743416874	2.6370933339345997	2.82395816439038	SCB	17.259755103316397	1.45356541787482	1.45356541787482
CPALL	8.3168715973838	1.48384200809295	1.48384200809295	SCC	4.683021813852109	1.24623793245445	1.2538614153382
CPF	25.652080069104098	3.1644538152052695	3.05845127611089	TASCO	58.245793363452606	2.73387567158041	2.95137300477178
CPN	16.7398510056753	1.35832525889927	1.35832525889927	TCAP	8.83621648430655	1.25013976176691	1.25013976176691
DELTA	6.612089873376539	2.3876049585544	2.3876049585544	TMB	5.76612160544923	1.8495684334071598	1.8800686891649698
DTAC	208.081746128544	3.8961890968552897	3.7174264735341	TOP	22.0253013917557	1.7444108294053602	1.7444108294053702
EGCO	15.050465632468802	1.1875826566385699	1.20048535812001	TPIPL	10.6986610149782	2.3253687912945398	2.7905349923157003
GLOW	6.24737610562749	1.7642093030222	1.75581001273765	TRUE	30.951640031162604	3.27086502586268	2.61592895967272
GPSC	2.3743190650515102	1.61000950177373	1.61000950177373	TTW	8.044429941551599	1.10972067907466	1.11779768707395
HMPRO	18.0299873924915	1.5263979556168201	1.5224076220698501	TU	8.65828903221027	2.2198647037494803	2.21995916361447
INTUCH	54.4554791334238	2.10045399948516	2.10045399948516	WHA	11.2164527491502	2.51648347771346	2.56150680987964
IRPC	23.4661981303641	1.71839185857574	1.85078082125751	-	-	-	-

Table 5.16 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. It gives better results than using EMD from closing price and closing change for closing change and tick change model, but gives higher error for closing price model. It still gives higher error results than the previous models for all based data.

Table 5.16 Average error for EMD of tick change based model.

Base	MAPE	MSE	R^2
Close	27.242295154774	1631.59033807538	-14.8436733221361
Change	2.0203760988903	9.26150819429554	0.88498023987191
Tick	2.0069281612238	6.11947219220167	0.88279500957534

5.1.5 Multi-class Prediction Model

Next, we are going to create multiple models based on some characteristic of a stock price. At any given time, a stock price has its own trend of price movement, which can be varied from time to time. So we construct a multi-class model based on a trend of stock price. We use the popular MACD for classifying a trend. Usually, MACD is used for identify Bearish-Bullish trend, and buy-sell signal.

(1) MACD crossing above the signal line is interpreted as buying signal.

(2) MACD crossing below the signal line is interpreted as selling signal.

(3) MACD crossing above the zero line is interpreted as bullish trend.

(4) MACD crossing below the zero line is interpreted as bearish trend.

From above signals, we divide the dataset into four classes:

(1) MACD above the signal line, and above the zero line.

(2) MACD above the signal line, but below the zero line.

(3) MACD below the signal line, but above the zero line.

(4) MACD below the signal line, and below the zero line.

Thus, the prediction model becomes:

$$close_n = f_{(i,close)}(close_{n-1})$$

$$change_n = f_{(i,change)}(change_{n-1})$$

$$tick_n = f_{(i,tick)}(tick_{n-1})$$

where $close_{n-1}$ is in i^{th} class and $i \in \{1, 2, 3, 4\}$.

Table 5.17 shows the results of this model with EPS-BSVR and Bessel kernel.

Table 5.18 shows the average error of EPS-BSVR with Bessel kernel from all stocks by based data. It gives a pretty good result, just a bit higher error than the volume model for all based data. But it requires a lot of data for training, so there is a stock, i.e., GPSC, that does not have enough records to use with this model. So this reult is an average result over 48 stocks.

Table 5.17 Prediction results for multi-class based model.

Stock	Close	Change	Tick	Stock	Close	Change	Tick
ADVANC	26.9945354369428	1.7440819632855702	1.7844231235018202	IVL	6.21142582869875	2.13390631301775	2.19139763994775
AOT	19.4093891333838	1.03773907008584	1.03827796731824	KBANK	2.1020395722236698	1.39382845506808	1.4086984146351
BA	1.11474350080051	1.04244472040619	1.0424447204062	KCE	20.1829014139464	1.44445084745866	1.45390676602722
BANPU	17.4809926063579	2.0115322576448698	2.0115322588211	KTB	1.18624084369696	1.15122124169238	1.15122124530782
BBL	3.02313478790753	1.22232783516157	1.22742143163391	LH	1.24708563378237	1.25442082366503	1.23998850267269
BCP	1.9267447302390501	1.3595983502008901	1.3595983502008901	MINT	14.2806376630983	1.6711566829436901	1.6819939216929598
BDMS	9.33202596116814	0.880699045061106	0.8806990450630949	MTLS	3.5656848833407904	0.9492205019861489	0.949220501986148
BEC	13.3136740405936	1.68723456974666	1.6797941164544001	PS	1.39036704455575	1.23519959511927	1.26227798498878
BH	2.32439720979834	1.39317650438794	1.4027405356931402	PTT	4.40283081651087	1.84146108608008	1.84274584780154
BLA	7.87229122370033	1.59277770969536	1.5927777096396498	PTTEP	12.0501371127646	2.24820095840083	2.2015983123239597
BTS	0.9543609684755939	0.928909425560279	0.9285550751383469	PTTGC	4.262546015952131	1.8242326657025	1.8242326657025
CBG	29.253438744439503	1.9926987695512302	1.9926987695512302	ROBINS	3.8186443517606703	1.44280572160401	1.44280572160401
CENTEL	6.69886158623406	1.55192424686963	1.54974835846529	SAWAD	2.0309239673109802	1.41632714878244	1.41632714878245
CK	4.73787206669396	1.3696617577901	1.39795179510814	SCB	3.9570781210693804	1.42293710682975	1.42293710682975
CPALL	2.076128038177	1.32544494828328	1.32544494828328	SCC	1.21303477478489	1.08739488257123	1.08896106552184
CPF	6.23208050268221	1.9801600938165	1.9834173552217602	TASCO	81.6551832278829	2.29156532410381	2.39811906453658
CPN	10.5979251341586	1.2247736583528301	1.2247736583528301	TCAP	5.046931035944549	1.16246095481338	1.16246095481338
DELTA	1.8048793203726199	1.71727962681091	1.71727962681091	TMB	1.50718634390483	1.36485591672463	1.36978900863283
DTAC	59.918749788757	3.1021786115102	3.09676153797062	TOP	20.025436492295	1.52098636724061	1.52098636724061
EGCO	8.28912578269712	0.894780637131108	0.8945814616959589	TPIPL	2.3564102653890298	1.63625328683937	1.6064961497726098
GLOW	1.42325901527908	1.28034646363883	1.28034646363883	TRUE	2.45805328638205	1.9753963114648598	1.9869093405342102
GPSC	0.0	0.0	0.0	TTW	0.989059550716934	0.821315645836264	0.817442707313807
HMPRO	4.10106547500228	1.46203680020543	1.45941180170106	TU	1.33115534476991	1.30898671768406	1.30898671768406
INTUCH	22.6891482244998	1.60605890945885	1.60605890945885	WHA	1.7803942033560598	1.5955181054879999	1.60515351102767
IRPC	9.38785331843362	1.29369201827949	1.3175765502858299	-	-	-	-

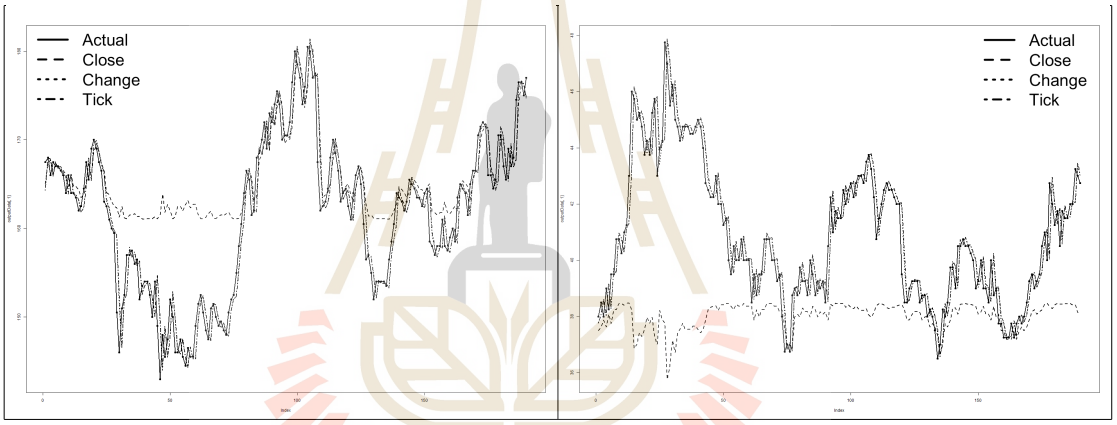

Figure 5.9 Graph of predicted results from two stocks for multi-class based model.

There are additional models that are not mentioned here, their models and results can be found in an Appendix section.

From all the results, we can improve the prediction result, but not by much. The reason is that SVRs tend to identify a predicted tube with acceptable error. And by the nature of stock price movement, it moves up and down from time to time. Or we can say its closing change moves between plus and minus values, i.e., mean-reversion. This can result in a large gap in a predicted tube, i.e., large error.

To improve the forecasted result further, we need to reduce this error. The solution we come up with is to use multi-SVR models, i.e., one prediction model

Table 5.18 Average error for multi-class based model.

Base	MAPE	MSE	R^2
Close	9.7918346748111	274.747430996931	-1.11875462871618
Change	1.4978262636261	4.22758392820576	0.934116994330382
Tick	1.5031035882874	4.25699288295389	0.933892403453394

for predicting plus line, and another model for predicting minus line.

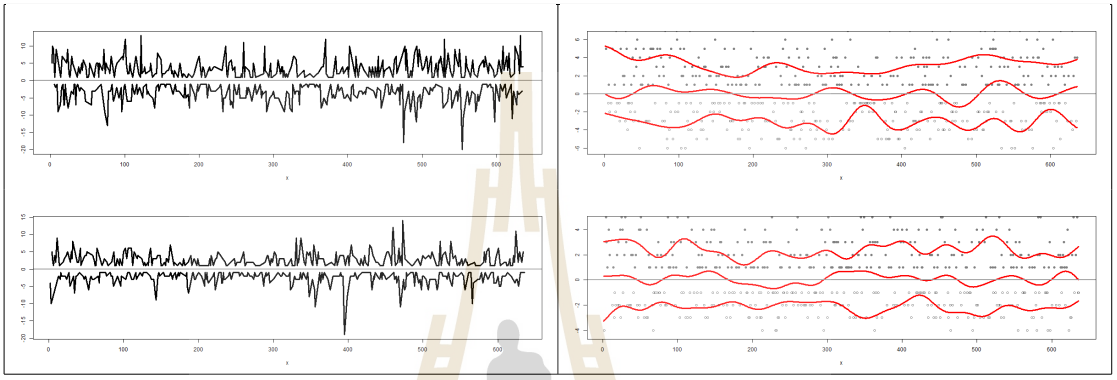

Figure 5.10 The left charts show the plus line and minus line of tick change. The right charts show the predicted lines, i.e., the middle line is the predicted line of tick change from a single price prediction model, the top and bottom lines is the predicted lines of plus tick change and minus tick change respectively.

Figure 5.10 illustrates charts from two stocks. The left charts show the plus line and minus line of tick change. The middle lines of the right charts show the predicted tick change; while the top and bottom lines show the predicted plus and minus tick change respectively. It is clear that the middle lines have larger error compare to those of plus and minus line. Table 5.19 shows the average MAPE from all stocks for these three predicted lines in the right charts by time series based model with EPS-BSVR and Bessel kernel. We calculated them with in-sample method, so the error is lower than the usual out-of-sample that we used for experiment.

It is clear from the results, that the error is reduced greatly when the predicted value is in the correct price movement direction. On the other hand, the error is also increased quite a lot for the incorrect direction. So the problem now is

Table 5.19 Average error for a single predicted line, predicted plus and minus lines.

Base	MAPE	Base	MAPE
Change	1.0973320108057	Tick	1.1157557296846
Plus Change (Correct Direction)	0.8261323258805	Plus Tick (Correct Direction)	0.8278777855592
Plus Change (Incorrect Direction)	3.0497039328437	Plus Tick (Incorrect Direction)	3.0801752844256
Minus Change (Correct Direction)	0.5411534369647	Minus Tick (Correct Direction)	0.5506562576777
Minus Change (Incorrect Direction)	2.5808688302544	Minus Tick (Incorrect Direction)	2.5574497029311

to determine the movement of stock price, which is also an interesting topic. The price movement is as important as (if not more) the actual predicted value. We are going to discuss the model for predicting price movement in the next section.

5.2 Direction Prediction Model

In this section, we focus on building a prediction model to forecast the movement of stock price. Since the price movement is just "move up" or "move down", so we do not plot its graph. We use a model similar to that of price prediction with an addition of direction as basis data.

First, we define price direction as:

$$direction_n = 1; close_n \geq close_{n-1}$$

$$direction_n = -1; otherwise$$

Clearly, for closing change:

$$direction_n = 1; change_n \geq 0$$

$$direction_n = -1; otherwise$$

and for tick change:

$$direction_n = 1; tick_n \geq 0$$

$$direction_n = -1; otherwise$$

All the results shown in this section are from EPS-BSVR with Laplace RBF kernl, since it gives the best result for direction prediction. The full list of results from all combinations of SVR type and kernel function can be found in the Appendix.

5.2.1 Time Series based Model

Firstly, we construct the prediction model that is similar to a time series model.

$$close_n = f_{close}(n)$$

$$change_n = f_{change}(n)$$

$$tick_n = f_{tick}(n)$$

$$direction_n = f_{direction}(n)$$

Then we use a predicted close, closing change, and tick change to calculate predicted direction. Table 5.20 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.21 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. The model from direction data gives the best accuracy, about 57.8%, and the rest give about 50% correctness.

Table 5.20 MAPE results for time series based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	43.1578947368421	44.7368421052632	44.2105263157895	43.684210526315795	IVL	48.421052631578895	57.368421052631604	57.368421052631604	41.0526315789474
AOT	61.0526315789474 05	45.78947368421 05	43.684210526315795	43.1578947368421	KBANK	45.2631578947368	43.684210526315795	43.684210526315795	43.684210526315795
BA	42.592592592592595	40.7407407407407	44.4444444444444	38.88888888888889	KCE	60.0	60.0	60.0	41.5789473684211
BANPU	41.5789473684211	58.421052631578995	58.421052631578995	40.0	KTB	39.4736842105263	41.5789473684211	42.105263157894704	39.4736842105263
BBL	44.7368421052632	44.7368421052632	45.2631578947368	44.7368421052632	LH	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789
BCP	49.4736842105263	61.57894736842 11	61.57894736842 11	40.5263157894737	MINT	57.894736842105296	44.2105263157895	43.684210526315795	42.105263157894704
BDMS	62.6315789473684	62.6315789473684	62.6315789473684	37.368421052631604	MTLS	51.020408163265294	51.020408163265294	51.020408163265294	51.020408163265294
BEC	45.2631578947368	54.736842105263 19	54.736842105263 19	49.4736842105263	PS	40.0	37.368421052631604	36.8421052631579	37.368421052631604
BH	50.5263157894737	50.5263157894737	50.5263157894737	50.5263157894737	PTT	42.631578947368396	58.9473684210526	58.9473684210526	41.0526315789474
BLA	50.0	54.2105263157895	54.2105263157895	45.7894736842105	PTTEP	47.3684210526316	52.631578947368396	52.631578947368396	42.631578947368396
BTS	45.7894736842105	59.4736842105263	59.4736842105263	42.105263157894704	PTTGC	46.842105263157904	53.1578947368421	53.1578947368421	47.3684210526316
CBG	68.0	32.0	32.0	32.0	ROBINS	56.3157894736842	62.1052631578947	62.1052631578947	38.4210526315789
CENTEL	58.421052631578995	45.26315789473 68	45.2631578947368	43.684210526315795	SAWAD	50.0	58.88888888888889	41.1111111111111	41.1111111111111
CK	58.421052631578995	40.5263157894737	40.0	38.9473684210526	SCB	41.0526315789474	58.9473684210526	58.9473684210526	39.4736842105263
CPALL	54.2105263157895	43.1578947368421	43.1578947368421	43.1578947368421	SCC	47.8947368421053	41.5789473684211	42.105263157894704	40.0
CPF	51.5789473684211 05	56.3157894736842	56.3157894736842	45.2631578947368	TASCO	53.6842105263158	46.3157894736842	46.3157894736842	46.3157894736842
CPN	51.5789473684211 05	61.57894736842 11	61.57894736842 11	39.4736842105263	TCAP	57.368421052631604	43.1578947368421	43.1578947368421	41.5789473684211
DELTA	52.631578947368396	54.736842105263 19	54.736842105263 19	44.7368421052632	TMB	46.842105263157904	56.3157894736842	45.2631578947368	43.684210526315795
DTAC	50.0	52.1052631578947	52.1052631578947	50.0	TOP	60.0	45.2631578947368	44.2105263157895	40.5263157894737
EGCO	60.0	40.5263157894737	38.9473684210526	38.9473684210526	TPIPL	56.3157894736842	55.7894736842105	55.7894736842105	44.2105263157895
GLOW	42.105263157894704	43.1578947368421	49.4736842105263	41.0526315789474	TRUE	45.7894736842105	54.736842105263 19	54.736842105263 19	45.2631578947368
GPSC	56.25	56.25	56.25	43.75	TTW	38.4210526315789	65.2631578947368	65.2631578947368	34.7368421052632
HMPRO	53.6842105263158	63.1578947368421	63.1578947368421	36.84210526315 79	TU	52.1052631578947	60.0	60.0	41.0526315789474
INTUCH	41.0526315789474	42.105263157894704	42.105263157894704	42.105263157894704	WHA	43.1578947368421	55.2631578947368	55.2631578947368	46.842105263157904
IRPC	56.842105263157904	56.842105263157904	56.842105263157904	43.1578947368421	-	-	-	-	-

Table 5.21 Average error for time series based model.

Base	MAPE
Close	50.1604694031808
Change	51.169818171928
Tick	50.6785105377632
Direction	42.2112113373813

5.2.2 Historical data based Model

Next, we use historical data to predict. We use a week of historical data, e.g.,the previous 5 days, and do prediction for a day ahead.

$$close_n = f_{close}(close_{n-1}, close_{n-2}, close_{n-3}, close_{n-4}, close_{n-5})$$

$$change_n = f_{change}(change_{n-1}, change_{n-2}, change_{n-3}, change_{n-4}, change_{n-5})$$

$$tick_n = f_{tick}(tick_{n-1}, tick_{n-2}, tick_{n-3}, tick_{n-4}, tick_{n-5})$$

$$direction_n = f_{direction}(direction_{n-1}, direction_{n-2}, direction_{n-3}, direction_{n-4}, direction_{n-5})$$

Table 5.22 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.23 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. The direction model gives the best accuracy, about

Table 5.22 MAPE results for historical data based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	44.7368421052632	42.631578947368396	43.1578947368421	43.684210526315795	IVL	56.842105263157904	55.7894736842105	51.052631578947405	46.3157894736842
AOT	58.421052631578995	50.5263157894737	46.842105263157904	40.52631578947 37	KBANK	46.3157894736842	48.9473684210526	46.842105263157904	46.842105263157904
BA	51.851851851851805	48.148148148148 1	42.592592592592595	38.88888888888889	KCE	61.052631578947405	50.5263157894737	51.052631578947405	43.1578947368421
BANPU	41.0526315789474	50.0	49.4736842105263	36.84210526315 79	KTB	46.3157894736842	38.9473684210526	40.0	40.52631578947 37
BBL	41.0526315789474	51.578947368421105	51.052631578947405	46.842105263157904	LH	43.684210526315795	48.9473684210526	54.2105263157895	39.4736842105263
BCP	53.1578947368421	44.7368421052632	44.7368421052632	40.0	MINT	59.4736842105263	48.421052631578895	50.0	41.5789473684211
BDMS	57.368421052631604	47.8947368421053	47.8947368421053	39.4736842105263	MTLS	55.1020408163265	53.0612244897959	53.0612244897959	53.0612244897959
BEC	45.2631578947368	55.7894736842105	56.3157894736842	51.578947368421105	PS	48.421052631578895	44.7368421052632	50.5263157894737	40.0
BH	53.1578947368421	53.1578947368421	51.578947368421105	45.7894736842105	PTT	42.631578947368396	51.052631578947405	50.5263157894737	47.8947368421053
BLA	49.4736842105263	48.421052631578895	48.421052631578895	44.2105263157895	PTTEP	45.7894736842105	58.9473684210526	51.578947368421105	53.1578947368421
BTS	50.0	50.5263157894737	46.3157894736842	40.0	PTTGC	54.2105263157895	50.5263157894737	50.5263157894737	50.0
CBG	70.0	44.0	44.0	40.0	ROBINS	50.5263157894737	49.4736842105263	50.0	42.105263157894704
CENTEL	56.8421052631579 04	46.3157894736842	46.3157894736842	38.4210526315789	SAWAD	38.88888888888889	45.5555555555556	45.5555555555556	42.22222222222222
CK	55.7894736842105	42.631578947368396	48.421052631578895	38.9473684210526	SCB	42.105263157894704	52.1052631578947	51.578947368421105	41.0526315789474
CPALL	53.6842105263158	46.3157894736842	44.2105263157895	41.57894736842 11	SCC	43.684210526315795	46.3157894736842	47.8947368421053	45.2631578947368
CPF	50.0	52.631578947368396	53.1578947368421	41.5789473684211	TASCO	53.6842105263158	45.2631578947368	49.4736842105263	48.421052631578895
CPN	57.368421052631604	52.1052631578947	50.5263157894737	48.9473684210526	TCAP	54.2105263157895	52.631578947368396	52.1052631578947	38.4210526315789
DELTA	51.578947368421105	46.3157894736842	46.842105263157904	50.0	TMB	54.73684210526319	49.4736842105263	50.5263157894737	44.2105263157895
DTAC	50.0	53.1578947368421	50.5263157894737	53.1578947368421	TOP	59.4736842105263	46.842105263157904	45.7894736842105	49.4736842105263
EGCO	55.2631578947368	47.36842105263 16	46.842105263157904	37.368421052631604	TPIPL	48.9473684210526	52.1052631578947	48.9473684210526	48.421052631578895
GLOW	48.9473684210526	46.3157894736842	44.2105263157895	46.842105263157904	TRUE	45.2631578947368	43.1578947368421	47.3684210526316	47.3684210526316
GPSC	62.5	62.5	62.5	43.75	TTW	39.4736842105263	48.421052631578895	48.421052631578895	34.7368421052632
HMPRO	57.894736842105296	54.73684210526319	56.3157894736842	36.84210526315 79	TU	53.6842105263158	42.105263157894704	42.105263157894704	41.0526315789474
INTUCH	41.0526315789474	45.2631578947368	42.631578947368396	44.2105263157895	WHA	54.2105263157895	46.842105263157904	45.7894736842105	54.2105263157895
IRPC	54.73684210526319	42.105263157894704	44.2105263157895	48.421052631578895	-	-	-	-	-

56%, but the time series model still gives a better correctness.

Table 5.23 Average error for historical data based model.

Base	MAPE
Close	51.2228924270508
Change	48.8851059459468
Tick	48.653574736972
Direction	44.0177490616727

From the results, the direction based model gives the best result.

5.2.3 Technical Analysis based Model

We use similar indicators as those for price prediction with one additional indicator:

(1) EMA from 10 days (2 weeks), 60 days (3 monthes), and 250 days (1 year)

(2) Volume

(3) Buy Volume, Sell Volume

(4) RSI (Relative Strength Index)

With EMA, the equations become:

$$close_n = f_{close}(close_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

$$change_n = f_{change}(change_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

$$direction_n = f_{direction}(direction_{n-1}, ema10_{n-1}, ema60_{n-1}, ema250_{n-1})$$

Table 5.24 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.24 MAPE results for EMA based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	43.1578947368421	50.0	43.684210526315795	43.684210526315795	IVL	54.73684210526319	50.5263157894737	50.5263157894737	49.47368421052631
AOT	60.52631578947369	46.842105263157904	43.1578947368421	38.9473684210526	KBANK	42.105263157894704	46.3157894736842	47.3684210526316	45.2631578947368
BA	35.185185185185205	37.037037037037	38.88888888888889	38.88888888888889	KCE	60.0	42.105263157894704	51.578947368421105	41.5789473684211
BANPU	41.0526315789474	51.578947368421105	42.105263157894704	40.0	KTB	40.5263157894737	40.5263157894737	38.9473684210526	40.0
BBL	45.2631578947368	42.631578947368396	43.1578947368421	44.7368421052632	LH	45.26315789473368	59.4736842105263	60.526315789473369	41.0526315789474
BCP	42.631578947368396	44.7368421052632	44.7368421052632	40.5263157894737	MINT	57.894736842105296	57.894736842105296	50.5263157894737	42.105263157894704
BDMS	61.5789473684211	63.6842105263158	63.6842105263158	54.21052631578895	MTLS	53.0612244897959	51.020408163265294	53.0612244897959	48.9795918367347
BEC	45.2631578947368	54.736842105263319	54.736842105263319	46.842105263157904	PS	37.368421052631604	37.368421052631604	40.0	36.3157894736842
BH	50.5263157894737	47.89473684210053	47.3684210526316	50.0	PTT	44.21052631578895	48.421052631578895	49.4736842105263	41.0526315789474
BLA	51.052631578947405	50.5263157894737	48.421052631578895	47.3684210526316	PTTEP	47.3684210526316	52.631578947368396	53.1578947368421	52.631578947368396
BTS	44.7368421052632	43.1578947368421	42.631578947368396	38.9473684210526	PTTGC	48.421052631578895	44.21052631578895	44.21052631578895	46.3157894736842
CBG	68.0	36.0	34.0	32.0	ROBINS	56.842105263157904	61.5789473684211	61.052631578947405	37.894736842105296
CENTEL	58.9473684210526	58.421052631578995	56.842105263157904	42.631578947368396	SAWAD	47.77777777777778	53.33333333333333	53.33333333333333	42.22222222222222
CK	58.9473684210526	45.78947368421105	43.1578947368421	38.9473684210526	SCB	40.52631578947737	51.052631578947405	50.5263157894737	42.105263157894704
CPALL	52.10526315789474	51.578947368421105	52.631578947368396	43.1578947368421	SCC	43.1578947368421	45.26315789473368	42.631578947368396	42.105263157894704
CPF	52.631578947368396	58.421052631578995	56.3157894736842	45.26315789473368	TASCO	53.68421052631158	46.3157894736842	46.3157894736842	46.3157894736842
CPN	56.3157894736842	52.631578947368396	52.631578947368396	42.105263157894704	TCAP	56.3157894736842	50.5263157894737	50.5263157894737	41.0526315789474
DELTA	50.5263157894737	43.1578947368421	43.684210526315795	45.26315789473368	TMB	48.421052631578895	45.7894736842105	45.26315789473368	43.684210526315795
DTAC	50.0	52.631578947368396	53.68421052631158	50.0	TOP	60.526315789473369	43.684210526315795	42.631578947368396	44.7368421052632
EGCO	58.421052631578995	54.21052631578895	54.736842105263319	40.52631578947737	TPIPL	46.842105263157904	45.26315789473368	44.736842105263232	41.5789473684211
GLOW	46.3157894736842	40.5263157894737	42.631578947368396	39.4736842105263	TRUE	47.36842105263316	46.3157894736842	47.89473684210053	43.1578947368421
GPSC	56.25	37.5	37.5	43.75	TTW	35.78947368421105	46.3157894736842	40.5263157894737	34.7368421052632
HMPRO	52.631578947368396	48.94736842105263	50.0	36.842105263157895	TU	49.4736842105263	43.1578947368421	42.631578947368396	40.0
INTUCH	41.5789473684211	42.631578947368396	42.105263157894704	42.105263157894704	WHA	46.3157894736842	46.3157894736842	46.842105263157904	44.7368421052632
IRPC	55.26315789473368	49.4736842105263	47.36842105263316	45.78947368421105	-	-	-	-	-

Table 5.25 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. Similar to the previous models, direction model gives the best accuracy, about 57.1%. It is slightly lower than the time series model of direction data.

Table 5.25 Average error for EMA based model.

Base	MAPE
Close	49.8552197224516
Change	48.1664068658851
Tick	47.6357524033602
Direction	42.8796706294405

With volume, the equations become:

$$close_n = f_{close}(close_{n-1}, vol_{n-1})$$

$$change_n = f_{change}(change_{n-1}, vol_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, vol_{n-1})$$

$$direction_n = f_{direction}(direction_{n-1}, vol_{n-1})$$

Table 5.26 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.26 MAPE results for volume based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	42.631578947368396	45.7894736842105	49.4736842105263	46.3157894736842	IVL	55.2631578947368	52.631578947368396	45.2631578947368	53.1578947368421
AOT	58.9473684210526	50.0	47.8947368421053	44.7368421052632	KBANK	45.2631578947368	47.8947368421053	45.7894736842105	44.7368421052632
BA	44.4444444444444	29.6296296296296	40.7407407407407	38.8888888888889	KCE	60.0	53.1578947368421	51.578947368421105	47.3684210526316
BANPU	40.5263157894737	47.3684210526316	48.421052631578895	40.0	KTB	50.0	51.052631578947405	51.578947368421105	41.0526315789474
BBL	43.1578947368421	49.4736842105263	53.1578947368421	43.684210526315795	LH	48.9473684210526	44.2105263157895	42.631578947368396	39.4736842105263
BCP	52.10526315789 47	51.052631578947405	51.052631578947405	37.368421052631604	MINT	59.4736842105263	51.052631578947405	45.2631578947368	38.4210526315789
BDMS	59.4736842105263	46.842105263157904	46.842105263157904	39.4736842105263	MTLS	57.142857142857096	46.9387755102041	46.9387755102041	51.020408163265294
BEC	46.3157894736842	44.7368421052632	45.78947368421105	48.421052631578895	PS	47.8947368421053	47.8947368421053	48.9473684210526	37.368421052631604
BH	55.7894736842105	42.631578947368396	46.3157894736842	44.2105263157895	PTT	42.631578947368396	46.3157894736842	45.7894736842105	47.8947368421053
BLA	49.4736842105263	50.5263157894737	50.5263157894737	50.5263157894737	PTTEP	47.3684210526316	51.578947368421105	53.1578947368421	53.1578947368421
BTS	52.10526315789 47	52.631578947368396	51.578947368421105	46.842105263157904	PTTGC	53.6842105263158	48.9473684210526	48.9473684210526	47.3684210526316
CBG	68.0	40.0	38.0	32.0	ROBINS	51.578947368421105	52.631578947368396	53.6842105263158	45.2631578947368
CENTEL	54.7368421052631 9	48.9473684210526	50.0	46.3157894736842	SAWAD	51.11111111111109	46.6666666666667	47.7777777777778	38.8888888888889
CK	56.3157894736842	48.421052631578895	49.4736842105263	39.4736842105263	SCB	44.7368421052632	55.2631578947368	55.2631578947368	54.7368421052631 9
CPALL	55.2631578947368	51.578947368421105	52.631578947368396	43.1578947368421	SCC	48.421052631578895	44.7368421052632	44.7368421052632	38.9473684210526
CPF	47.3684210526316	57.368421052631604	50.5263157894737	48.9473684210526	TASCO	55.7894736842105	45.2631578947368	43.684210526315795	46.842105263157904
CPN	58.421052631578995	53.6842105263158	53.6842105263158	42.631578947368396	TCAP	52.1052631578947	46.842105263157904	47.3684210526316	37.368421052631604
DELTA	54.7368421052631 9	45.2631578947368	45.2631578947368	47.3684210526316	TMB	54.2105263157895	52.1052631578947	53.6842105263158	41.5789473684211
DTAC	49.4736842105263	47.8947368421053	51.578947368421105	52.631578947368396	TOP	59.4736842105263	46.842105263157904	46.842105263157904	43.684210526315795
EGCO	56.842105263157904	50.0	48.421052631578895	43.1578947368421	TPIPL	49.4736842105263	53.6842105263158	55.2631578947368	46.842105263157904
GLOW	52.1052631578947	49.4736842105263	48.421052631578895	40.5263157894737	TRUE	48.421052631578895	51.052631578947405	49.4736842105263	44.2105263157895
GPSC	43.75	56.25	50.0	50.0	TTW	52.1052631578947	51.052631578947405	50.0	42.631578947368396
HMPRO	55.2631578947368	50.0	52.1052631578947	39.4736842105263	TU	50.5263157894737	50.0	49.4736842105263	45.2631578947368
INTUCH	44.7368421052632	47.3684210526316	47.3684210526316	41.0526315789474	WHA	55.7894736842105	51.052631578947405	50.5263157894737	47.8947368421053
IRPC	56.3157894736842	56.3157894736842	53.6842105263158	45.7894736842105	-	-	-	-	-

Table 5.27 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. Unlike the price prediction which gives the best result, this model only gives a fair result for direction prediction.

Table 5.27 Average error for volume based model.

Base	MAPE
Close	51.830848379452
Change	49.0227887908953
Tick	48.9105140564401
Direction	44.2482981019117

With buy-sell volume, the equations become:

$$close_n = f_{close}(close_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$change_n = f_{change}(change_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$direction_n = f_{direction}(direction_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

Table 5.28 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.28 MAPE results for buy-sell volume based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	44.7368421052632	53.1578947368421	52.631578947368396	45.2631578947368	IVL	55.2631578947368	54.73684210526319	49.4736842105263	53.1578947368421
AOT	60.0	45.2631578947368	47.3684210526316	44.2105263157895	KBANK	45.7894736842105	46.3157894736842	44.2105263157895	42.105263157894704
BA	53.703703703703695	31.4814814814815	35.185185185185205	31.4814814814815	KCE	61.052631578947405	54.2105263157895	52.1052631578947	46.3157894736842
BANPU	41.5789473684211	53.6842105263158	53.1578947368421	40.0	KTB	46.3157894736842	50.0	50.0	40.0
BBL	43.684210526315795	45.2631578947368	44.7368421052632	40.5263157894737	LH	51.578947368421105	46.842105263157904	50.0	42.10526315789470
BCP	54.2105263157895	46.842105263157904	46.842105263157904	42.631578947368396	MINT	58.9473684210526	47.3684210526316	46.842105263157904	43.1578947368421
BDMS	57.894736842105296	44.2105263157895	46.3157894736842	39.4736842105263	MTLS	57.142857142857096	51.020408163265294	46.9387755102041	51.020408163265294
BEC	47.8947368421053	51.578947368421105	50.5263157894737	47.8947368421053	PS	55.7894736842105	50.5263157894737	51.578947368421105	37.894736842105296
BH	54.73684210526319	48.421052631578895	50.0	53.6842105263158	PTT	43.1578947368421	49.4736842105263	49.4736842105263	48.9473684210526
BLA	47.8947368421053	49.4736842105263	45.7894736842105	52.1052631578947	PTTEP	49.4736842105263	56.842105263157904	59.4736842105263	58.421052631578995
BTS	53.6842105263158	48.9473684210526	48.9473684210526	43.684210526315795	PTTGC	56.8421052631579 04	53.1578947368421	53.6842105263158	45.2631578947368
CBG	70.0	42.0	42.0	32.0	ROBINS	51.578947368421105	55.2631578947368	55.2631578947368	44.7368421052632
CENTEL	57.368421052631604	47.8947368421053	46.3157894736842	43.1578947368421	SAWAD	44.44444444444444	54.44444444444439	55.5555555555556	41.11111111111111
CK	54.2105263157895	50.0	48.9473684210526	40.5263157894737	SCB	42.105263157894704	55.2631578947368	55.7894736842105	50.5263157894737
CPALL	56.842105263157904	53.6842105263158	53.6842105263158	43.1578947368421	SCC	51.052631578947405	47.3684210526316	48.421052631578895	42.105263157894704
CPF	46.3157894736842	53.1578947368421	51.052631578947405	41.052631578947 4	TASCO	52.1052631578947	46.3157894736842	51.578947368421105	47.8947368421053
CPN	61.5789473684211	53.6842105263158	52.1052631578947	41.052631578947 4	TCAP	54.73684210526319	46.3157894736842	45.7894736842105	35.7894736842105
DELTA	54.2105263157895	52.1052631578947	52.1052631578947	50.5263157894737	TMB	55.7894736842105	46.842105263157904	47.8947368421053	43.684210526315795
DTAC	51.578947368421105	53.1578947368421	53.1578947368421	48.421052631578895	TOP	60.0	43.684210526315795	42.631578947368396	40.5263157894737
EGCO	56.3157894736842	43.684210526315795	44.2105263157895	39.4736842105263	TPIPL	57.368421052631604	52.631578947368396	53.1578947368421	41.57894736842 11
GLOW	50.0	52.1052631578947	50.5263157894737	42.631578947368396	TRUE	48.421052631578895	48.9473684210526	51.052631578947405	44.2105263157895
GPSC	56.25	56.25	56.25	43.75	TTW	57.368421052631604	44.7368421052632	43.1578947368421	33.6842105263158
HMPRO	56.842105263157904	57.894736842105296	57.368421052631604	35.7894736842105	TU	51.052631578947405	45.7894736842105	45.7894736842105	38.9473684210526
INTUCH	42.105263157894704	54.2105263157895	52.631578947368396	43.684210526315795	WHA	59.4736842105263	45.7894736842105	47.3684210526316	43.684210526315795
IRPC	54.73684210526319	51.578947368421105	50.5263157894737	42.105263157894704	-	-	-	-	-

Table 5.29 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. The direction model gives a better accuracy comparing to using a combined trading volume, but it gives lower accuracy than the time series with direction data.

Table 5.29 Average error for buy-sell volume based model.

Base	MAPE
Close	52.9637799146392
Change	49.6656609534851
Tick	49.5839536076992
Direction	43.3704586620422

With RSI, the equations become:

$$close_n = f_{close}(close_{n-1}, rsi_{n-1})$$

$$change_n = f_{change}(change_{n-1}, rsi_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, rsi_{n-1})$$

$$direction_n = f_{direction}(direction_{n-1}, rsi_{n-1})$$

Table 5.30 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.30 MAPE results for RSI based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	43.1578947368421	52.6315789473683 96	55.2631578947368	47.8947368421053	IVL	54.7368421052631 9	52.6315789473683 96	51.0526315789474 05	40.0
AOT	56.8421052631579 04	44.2105263157895	50.0	41.0526315789474	KBANK	45.2631578947368	46.8421052631579 04	43.6842105263157 95	46.3157894736842
BA	50.0	37.0370370370370 37	42.5925925925925 95	35.1851851851852 05	KCE	60.0	51.5789473684211 05	51.5789473684211 05	43.6842105263157 95
BANPU	38.9473684210526	55.2631578947368	54.7368421052631 9	51.5789473684211 05	KTB	46.8421052631579 04	50.5263157894737	52.6315789473683 96	42.1052631578947 04
BBL	43.1578947368421	45.2631578947368	45.7894736842105	46.8421052631579 04	LH	48.9473684210526	53.1578947368421	54.2105263157895	44.2105263157895
BCP	54.7368421052631 9	46.3157894736842	44.2105263157895	41.0526315789474	MINT	56.8421052631579 04	48.4210526315789 5	49.4736842105263	47.3684210526316
BDMS	56.8421052631579 04	47.3684210526316	48.4210526315789 5	40.5263157894737	MTLS	55.1020408163265	44.8975918367347	46.9387755102041	48.9795918367347
BEC	45.2631578947368	50.5263157894737	47.8947368421053	47.8947368421053	PS	55.7894736842105	47.8947368421053	52.6315789473683 96	40.5263157894737
BH	53.1578947368421	44.2105263157895	47.3684210526316	45.7894736842105	PTT	42.1052631578947 04	49.4736842105263	48.9473684210526	40.5263157894737
BLA	48.4210526315789 5	47.8947368421053	47.8947368421053	49.4736842105263	PTTEP	46.3157894736842	55.2631578947368	55.7894736842105	50.5263157894737
BTS	53.6842105263158	53.6842105263158	50.0	43.1578947368421	PTTGC	52.6315789473683 96	50.0	50.5263157894737	46.8421052631579 04
CBG	68.0	52.0	50.0	48.0	ROBINS	53.1578947368421	53.1578947368421	53.1578947368421	45.2631578947368
CENTEL	55.7894736842105	56.3157894736842	56.3157894736842	40.5263157894737	SAWAD	53.3333333333333	47.7777777777778	46.6666666666667	53.3333333333333
CK	54.7368421052631 9	51.5789473684211 05	50.0	39.4736842105263	SCB	40.0	46.8421052631579 04	46.8421052631579 04	40.0
CPALL	55.2631578947368	56.3157894736842	54.7368421052631 9	40.5263157894737	SCC	45.2631578947368	46.8421052631579 04	47.3684210526316	41.5789473684211
CPF	50.0	53.6842105263158	48.4210526315789 5	47.3684210526316	TASCO	53.1578947368421	48.4210526315789 5	50.0	48.4210526315789 5
CPN	57.8947368421052 96	44.7368421052632	43.6842105263157 95	37.3684210526316 04	TCAP	53.6842105263158	42.6315789473683 96	42.6315789473683 96	37.3684210526316 04
DELTA	56.3157894736842	52.1052631578947	51.5789473684211 05	47.8947368421053	TMB	57.3684210526316 04	56.8421052631579 04	56.3157894736842	42.1052631578947 04
DTAC	49.4736842105263	46.8421052631579 04	47.3684210526316	47.8947368421053	TOP	60.0	46.3157894736842	46.8421052631579 04	40.5263157894737
EGCO	54.7368421052631 9	51.5789473684211 05	50.0	41.0526315789474	TPIPL	53.1578947368421	55.2631578947368	53.6842105263158	44.2105263157895
GLOW	44.7368421052632	44.7368421052632	46.8421052631579 04	44.7368421052632	TRUE	48.9473684210526	49.4736842105263	50.5263157894737	43.1578947368421
GPSC	56.25	56.25	50.0	43.75	TTW	42.1052631578947 04	45.2631578947368	46.8421052631579 04	36.8421052631579
HMPRO	61.0526315789474 05	47.3684210526316	48.4210526315789 5	40.0	TU	52.6315789473683 96	53.6842105263158	53.1578947368421	44.2105263157895
INTUCH	41.0526315789474	48.4210526315789 5	48.9473684210526	45.2631578947368	WHA	53.6842105263158	47.3684210526316	47.3684210526316	39.4736842105263
IRPC	54.7368421052631 9	45.7894736842105	48.9473684210526	49.4736842105263	-	-	-	-	-

Table 5.31 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. This model gives about the same result as Buy-Sell Volume model with slight higher or lower accuracy for each based data.

Table 5.31 Average error for RSI based model.

Base	MAPE
Close	51.7411623081026
Change	49.4428493082398
Tick	49.5572101617828
Direction	43.9051708880234

5.2.4 EMD based Model

$$close_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$direction_n = f_{direction}(direction_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.32 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.32 MAPE results for EMD of closing price based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	44.7368421052632	43.684210526315795	43.684210526315795	43.68421052631579	IVL	50.52631578947	50.0	50.0	40.52631578947
AOT	61.052631578947405	51.052631578947405	53.1578947368421	55.78947368421	KBANK	43.1578947368421	45.26315789473	44.7368421052632	46.3157894736842
BA	48.1481481481481	38.88888888888889	38.88888888888889	38.88888888888889	KCE	59.47368421052	46.3157894736842	45.26315789473	42.631578947368396
BANPU	41.0526315789474	58.421052631578995	58.421052631578995	59.47368421052	KTB	39.47368421052	46.842105263157904	46.3157894736842	38.9473684210526
BBL	40.52631578947	49.47368421052	46.842105263157904	51.052631578947405	LH	50.52631578947	61.57894736842	60.526315789473	53.6842105263158
BCP	46.842105263157904	50.52631578947	50.52631578947	43.684210526315795	MINT	58.94736842105	54.736842105263	53.68421052631	50.0
BDMS	62.1052631578947	58.94736842105	58.94736842105	53.68421052631	MTLS	48.9795918367347	53.06122448979	53.06122448979	46.9387755102041
BEC	45.2631578947368	55.26315789473	55.26315789473	54.2105263157895	PS	42.631578947368	45.78947368421	45.26315789473	40.0
BH	49.4736842105263	44.2105263157895	45.26315789473	48.94736842105	PTT	41.05263157894	59.47368421052	58.94736842105	56.3157894736842
BLA	45.2631578947368	50.52631578947	51.052631578947405	50.52631578947	PTTEP	47.36842105263	52.10526315789	52.631578947368	52.10526315789
BTS	55.7894736842105	59.47368421052	59.47368421052	53.68421052631	PTTGC	52.10526315789	50.52631578947	50.52631578947	46.842105263157904
CBG	68.0	60.0	62.0	62.0	ROBINS	53.68421052631	61.57894736842	61.57894736842	54.2105263157895
CENTEL	59.4736842105263	58.94736842105	58.94736842105	53.68421052631	SAWAD	44.44444444444	38.88888888888	38.88888888888	41.11111111111
CK	50.52631578947	55.78947368421	54.2105263157895	46.3157894736842	SCB	43.1578947368	58.421052631578995	58.421052631578995	54.2105263157895
CPALL	53.68421052631	53.1578947368421	53.68421052631	50.0	SCC	47.8947368421053	54.736842105263	55.26315789473	55.7894736842105
CPF	45.2631578947368	52.10526315789	50.0	51.578947368421105	TASCO	43.684210526315795	46.3157894736842	39.4736842105263	40.5263157894737
CPN	49.4736842105263	53.68421052631	53.68421052631	51.052631578947405	TCAP	55.78947368421	51.578947368421105	51.578947368421105	47.3684210526316
DELTA	54.7368421052632	54.2105263157895	54.2105263157895	50.0	TMB	44.7368421052632	54.2105263157895	54.2105263157895	43.1578947368421
DTAC	50.0	48.947368421052	48.94736842105	47.8947368421053	TOP	60.0	58.421052631578995	58.421052631578995	56.842105263157904
EGCO	54.2105263157895	49.47368421052	48.94736842105	40.5263157894737	TPIPL	55.26315789473	50.52631578947	50.52631578947	43.1578947368421
GLOW	53.68421052631	45.78947368421	46.842105263157904	42.631578947368396	TRUE	53.1578947368421	57.368421052631604	58.421052631578995	56.842105263157904
GPSC	43.75	43.75	43.75	50.0	TTW	47.3684210526316	43.1578947368421	44.2105263157895	35.26315789473
HMPRO	56.3157894736842	58.94736842105	59.47368421052	50.0	TU	57.894736842105296	59.47368421052	59.47368421052	56.3157894736842
INTUCH	42.631578947368396	45.26315789473	45.26315789473	40.52631578947	WHA	46.842105263157904	47.36842105263	47.36842105263	46.3157894736842
IRPC	54.73684210526319	54.2105263157895	55.26315789473	50.0	-	-	-	-	-

Table 5.33 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. This gives about 50% accuracy for all based data.

Table 5.33 Average error for EMD of closing price based model.

Base	MAPE
Close	50.3041047305663
Change	51.8874232471363
Tick	51.7456402181352
Direction	48.6786645915079

$$change_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$direction_n = f_{direction}(direction_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.34 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.34 MAPE results for EMD of closing change based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	43.684210526315795	46.3157894736842	45.2631578947368	46.842105263157904	IVL	52.1052631578947	50.0	50.0	43.684210526315795
AOT	60.52631578947369	54.2105263157895	53.6842105263158	53.1578947368421	KBANK	46.842105263157904	53.6842105263158	51.052631578947405	47.8947368421053
BA	59.2592592592593	59.2592592592593	55.55555555555556	38.88888888888889	KCE	60.52631578947369	57.894736842105296	58.421052631578995	50.5263157894737
BANPU	41.5789473684211	56.3157894736842	55.7894736842105	46.842105263157904	KTB	47.8947368421053	43.684210526315795	43.1578947368421	40.0
BBL	45.2631578947368	47.8947368421053	48.421052631578895	46.842105263157904	LH	47.3684210526316	45.2631578947368	45.7894736842105	42.105263157894704
BCP	44.2105263157895	46.842105263157904	46.3157894736842	39.4736842105263	MINT	60.0	43.684210526315795	43.684210526315795	42.631578947368396
BDMS	60.0	36.8421052631579	36.8421052631579	35.7894736842105	MTLS	51.020408163265294	65.3061224489796	65.3061224489796	59.1836734693878
BEC	45.2631578947368	43.684210526315795	43.684210526315795	44.2105263157895	PS	52.631578947368396	36.3157894736842	37.368421052631604	36.3157894736842
BH	53.6842105263158	52.1052631578947	54.2105263157895	56.3157894736842	PTT	44.7368421052632	55.7894736842105	55.7894736842105	56.3157894736842
BLA	46.842105263157904	43.1578947368421	42.631578947368396	47.3684210526316	PTTEP	46.842105263157904	51.578947368421105	51.578947368421105	52.631578947368396
BTS	48.421052631578895	36.3157894736842	36.3157894736842	40.5263157894737	PTTGC	50.5263157894737	50.5263157894737	50.0	47.3684210526316
CBG	70.0	46.0	42.0	30.0	ROBINS	50.0	48.421052631578895	48.421052631578895	48.9473684210526
CENTEL	60.0	54.73684210526319	55.2631578947368	51.052631578947405	SAWAD	46.66666666666667	53.3333333333333	53.3333333333333	54.44444444444439
CK	49.4736842105263	43.1578947368421	40.5263157894737	38.9473684210526	SCB	40.52631578947737	56.842105263157904	57.368421052631604	60.52631578947369
CPALL	56.3157894736842	48.421052631578895	48.421052631578895	42.105263157894704	SCC	51.052631578947405	46.842105263157904	47.8947368421053	42.105263157894704
CPF	45.2631578947368	53.1578947368421	53.6842105263158	53.1578947368421	TASCO	56.842105263157904	48.9473684210526	51.578947368421105	43.1578947368421
CPN	54.73684210526319	41.05263157894474	41.05263157894474	38.9473684210526	TCAP	53.6842105263158	46.3157894736842	45.2631578947368	41.5789473684211
DELTA	59.4736842105263	53.1578947368421	53.1578947368421	48.9473684210526	TMB	48.421052631578895	55.2631578947368	55.7894736842105	43.684210526315795
DTAC	50.0	47.3684210526316	47.8947368421053	50.52631578947737	TOP	59.4736842105263	42.105263157894704	42.105263157894704	44.2105263157895
EGCO	58.421052631578995	54.2105263157895	54.2105263157895	48.421052631578895	TPIPL	58.421052631578995	40.5263157894737	42.631578947368396	42.105263157894704
GLOW	47.3684210526316	46.842105263157904	47.3684210526316	46.3157894736842	TRUE	48.421052631578895	46.3157894736842	45.7894736842105	45.7894736842105
GPSC	56.25	56.25	56.25	43.75	TTW	43.1578947368421	45.7894736842105	45.2631578947368	37.894736842105296
HMPRO	53.1578947368421	55.7894736842105	55.2631578947368	44.7368421052632	TU	54.7368421052631	56.3157894736842	56.3157894736842	53.6842105263158
INTUCH	42.105263157894704	53.6842105263158	53.1578947368421	52.631578947368396	WHA	43.684210526315795	45.2631578947368	46.842105263157904	44.7368421052632
IRPC	58.9473684210526	54.2105263157895	54.2105263157895	50.5263157894737	-	-	-	-	-

Table 5.35 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. It shows a little improvement over using EMD from closing price, but it still gives higher error than the time series with direction data.

Table 5.35 Average error for EMD of closing change based model.

Base	MAPE
Close	51.5475084293176
Change	49.3263432715251
Tick	49.2228305213958
Direction	46.0784888606356

$$tick_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$direction_n = f_{direction}(direction_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.36 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.36 MAPE results for EMD of tick change based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	44.7368421052632	41.57894736842111	41.05263157894737	40.52631578947368	IVL	50.0	52.10526315789474	52.63157894736842	45.26315789473684
AOT	61.052631578947405	56.31578947368421	56.31578947368421	57.368421052631604	KBANK	42.631578947368396	49.47368421052632	48.421052631578955	50.0
BA	48.148148148148	59.25925925925926	57.407407407407405	38.888888888888886	KCE	60.0	56.31578947368421	56.842105263157904	48.421052631578955
BANPU	40.52631578947368	58.94736842105263	58.421052631578955	57.894736842105296	KTB	46.842105263157904	40.0	40.0	40.52631578947368
BBL	46.31578947368421	46.842105263157904	46.842105263157904	45.78947368421053	LH	46.31578947368421	44.73684210526316	45.26315789473684	41.05263157894737
BCP	44.7368421052632	46.842105263157904	46.31578947368421	39.47368421052632	MINT	57.368421052631604	44.21052631578947	44.21052631578947	43.15789473684211
BDMS	58.94736842105263	36.31578947368421	36.842105263157896	35.26315789473684	MTLS	51.020408163265294	65.30612244897959	65.30612244897959	59.18367346938776
BEC	45.26315789473684	43.684210526315795	43.684210526315795	44.21052631578947	PS	49.47368421052632	36.842105263157896	36.31578947368421	36.31578947368421
BH	53.15789473684211	53.684210526315795	54.21052631578947	52.10526315789474	PTT	43.684210526315795	53.15789473684211	53.15789473684211	55.78947368421053
BLA	48.421052631578955	42.631578947368396	42.631578947368396	45.26315789473684	PTTEP	47.89473684210526	52.10526315789474	51.052631578947405	54.21052631578947
BTS	44.7368421052632	34.73684210526316	34.73684210526316	38.421052631578955	PTTGC	51.052631578947405	47.89473684210526	48.94736842105263	47.36842105263158
CBG	70.0	42.0	46.0	32.0	ROBINS	50.0	48.421052631578955	48.421052631578955	48.94736842105263
CENTEL	58.421052631578955	58.421052631578955	58.421052631578955	57.368421052631604	SAWAD	50.0	52.22222222222222	52.22222222222222	51.11111111111109
CK	58.94736842105263	40.0	40.0	38.94736842105263	SCB	40.52631578947368	56.842105263157904	57.368421052631604	60.526315789473684
CPALL	56.31578947368421	48.421052631578955	47.89473684210526	41.05263157894737	SCC	47.89473684210526	50.0	50.52631578947368	44.73684210526316
CPF	53.15789473684211	55.78947368421053	56.842105263157904	56.842105263157904	TASCO	52.10526315789474	47.89473684210526	51.052631578947405	47.89473684210526
CPN	54.73684210526316	40.52631578947368	40.52631578947368	38.94736842105263	TCAP	53.684210526315795	47.36842105263158	46.842105263157904	41.57894736842105
DELTA	59.47368421052632	53.15789473684211	53.15789473684211	48.94736842105263	TMB	50.0	44.21052631578947	44.21052631578947	42.631578947368396
DTAC	50.0	50.0	50.0	48.421052631578955	TOP	59.47368421052632	42.105263157894704	42.105263157894704	44.21052631578947
EGCO	58.421052631578955	53.684210526315795	53.684210526315795	49.47368421052632	TPIPL	52.631578947368396	41.57894736842105	43.15789473684211	43.684210526315795
GLOW	48.94736842105263	46.31578947368421	46.31578947368421	48.421052631578955	TRUE	48.421052631578955	47.36842105263158	46.842105263157904	45.26315789473684
GPSC	56.25	56.25	56.25	50.0	TTW	41.05263157894737	41.57894736842105	43.684210526315795	34.21052631578947
HMPRO	51.578947368421105	58.421052631578955	58.94736842105263	51.052631578947405	TU	54.21052631578947	55.78947368421053	55.78947368421053	52.631578947368396
INTUCH	42.105263157894704	53.15789473684211	52.631578947368396	52.631578947368396	WHA	49.47368421052632	44.21052631578947	44.21052631578947	44.73684210526316
IRPC	58.94736842105263	52.10526315789474	51.578947368421105	46.842105263157904	-	-	-	-	-

Table 5.37 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. It gives about the same accuracy as using EMD from closing change.

Table 5.37 Average error for EMD of tick change based model.

Base	MAPE
Close	51.2061789150557
Change	48.7923893390749
Tick	48.9651227599287
Direction	46.522545430632

$$direction_n = \sum_{i=1}^{m} imf_{(i,n)} + r_n$$

$$close_n = f_{close}(close_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$change_n = f_{change}(change_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$tick_n = f_{tick}(tick_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

$$direction_n = f_{direction}(direction_{n-1}, imf_{(1,n-1)}, imf_{(2,n-1)}, ..., imf_{(m,n-1)})$$

Table 5.38 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.38 MAPE results for EMD of direction based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	42.631578947368396	45.2631578947368	47.36842105263 16	43.684210526315795	IVL	58.421052631578995	54.73684210526319	54.73684210526319	42.631578947368396
AOT	58.9473684210526	48.9473684210526	55.2631578947368	38.9473684210526	KBANK	47.8947368421053	46.84210526315 7904	48.9473684210526	45.2631578947368
BA	42.592592592592595	38.88888888888889	50.0	38.88888888888889	KCE	61.57894736842 11	52.1052631578947	50.5263157894737	40.0
BANPU	41.5789473684211	48.421052631578995	46.31578947368 42	41.57894736842 11	KTB	48.9473684210526	48.421052631578895	48.421052631578895	39.4736842105263
BBL	45.7894736842105	42.105263157894704	39.4736842105263	44.7368421052632	LH	54.2105263157895	55.2631578947368	53.6842105263158	38.4210526315789
BCP	51.578947368421105	51.052631578947405	51.052631578947405	38.4210526315789	MINT	57.894736842105296	44.2105263157895	43.684210526315795	42.105263157894704
BDMS	60.0	53.6842105263158	53.6842105263158	37.368421052631604	MTLS	55.102040816326 5	53.0612244897959	53.0612244897959	48.9795918367347
BEC	45.2631578947368	54.2105263157895	53.6842105263158	45.2631578947368	PS	56.3157894736842	50.0	52.631578947368396	36.3157894736842
BH	55.2631578947368	46.3157894736842	51.578947368421105	49.4736842105263	PTT	41.0526315789474	53.1578947368421	52.631578947368396	41.0526315789474
BLA	49.4736842105263	52.1052631578947	52.1052631578947	45.7894736842105	PTTEP	46.842105263157904	48.421052631578895	51.052631578947405	54.2105263157895
BTS	54.2105263157895	46.842105263157904	49.4736842105263	40.5263157894737	PTTGC	53.6842105263158	52.1052631578947	52.1052631578947	46.842105263157904
CBG	70.0	44.0	44.0	32.0	ROBINS	51.052631578947405	60.0	60.0	37.894736842105296
CENTEL	54.7368421052631 9	49.4736842105263	49.4736842105263	41.5789473684211	SAWAD	53.3333333333333	61.11111111111111	61.11111111111111	41.11111111111111
CK	55.2631578947368	45.7894736842105	44.7368421052632	38.9473684210526	SCB	45.7894736842105	52.631578947368396	52.631578947368396	41.0526315789474
CPALL	55.7894736842105	53.1578947368421	53.1578947368421	43.1578947368421	SCC	43.684210526315795	48.421052631578895	45.7894736842105	40.0
CPF	50.0	53.6842105263158	52.1052631578947	43.684210526315795	TASCO	53.6842105263158	50.5263157894737	46.842105263157904	46.3157894736842
CPN	57.894736842105296	57.894736842105296	57.894736842105296	38.4210526315789	TCAP	55.7894736842105	47.8947368421053	47.8947368421053	38.4210526315789
DELTA	56.842105263157904	45.7894736842105	45.7894736842105	45.2631578947368	TMB	55.2631578947368	48.421052631578895	48.421052631578895	43.684210526315795
DTAC	50.0	50.52631578947 37	48.9473684210526	50.0	TOP	60.526315789473 69	44.7368421052632	44.7368421052632	40.0
EGCO	54.2105263157895	53.1578947368421	53.1578947368421	38.9473684210526	TPIPL	49.4736842105263	42.631578947368396	38.9473684210526	44.2105263157895
GLOW	50.0	50.0	50.0	41.5789473684211	TRUE	46.842105263157904	50.5263157894737	44.7368421052632	45.2631578947368
GPSC	50.0	43.75	62.5	43.75	TTW	41.0526315789474	42.105263157894704	40.5263157894737	34.7368421052632
HMPRO	57.894736842105296	49.4736842105263	48.421052631578895	36.8421052631579	TU	58.9473684210526	45.2631578947368	45.2631578947368	40.0
INTUCH	42.105263157894704	43.1578947368421	43.1578947368421	42.105263157894704	WHA	60.0	49.4736842105263	49.4736842105263	44.7368421052632
IRPC	54.2105263157895	51.052631578947405	51.052631578947405	43.1578947368421	-	-	-	-	-

Table 5.39 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. This model gives a pretty good result. Its direction model gives the best accuracy so far, about 58%.

Table 5.39 Average error for EMD of direction based model.

Base	MAPE
Close	52.3195825650943
Change	49.4043107038734
Tick	49.8418629177414
Direction	41.9762215305026

From the results, EMD of direction based model gives the best result so far.

5.2.5 Multi-class Prediction Model

Similar to price prediction model, we divide the dataset into four classes based on MACD signal:

(1) MACD above the signal line, and above the zero line.

(2) MACD above the signal line, but below the zero line.

(3) MACD below the signal line, but above the zero line.

(4) MACD below the signal line, and below the zero line.

$$close_n = f_{(i,close)}(close_{n-1})$$

$$change_n = f_{(i,change)}(change_{n-1})$$

$$tick_n = f_{(i,tick)}(tick_{n-1})$$

$$direction_n = f_{(i,direction)}(direction_{n-1})$$

where $close_{n-1}$ is in i^{th} class and $i \in \{1, 2, 3, 4\}$.

Table 5.40 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.40 MAPE results for multi-class based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	43.684210526315795	49.4736842105263	51.052631578947405	43.684210526315795	IVL	52.1052631578947	54.73684210526319	56.3157894736842	42.631578947368396
AOT	61.052631578947405	43.684210526315795	43.1578947368421	38.9473684210526	KBANK	45.7894736842105	46.3157894736842	50.5263157894737	45.26315789473 68
BA	48.1481481481481	44.4444444444444	35.185185185185205	38.8888888888889	KCE	60.52631578947369	50.5263157894737	48.421052631578895	40.0
BANPU	42.631578947368396	41.0526315789474	41.0526315789474	41.5789473684211	KTB	49.4736842105263	42.105263157894704	42.105263157894704	39.4736842105263
BBL	44.2105263157895	41.5789473684211	42.631578947368396	44.7368421052632	LH	40.0	52.1052631578947	54.73684210526319	38.4210526315789
BCP	56.842105263157904	44.7368421052632	44.7368421052632	38.4210526315789	MINT	57.894736842105296	44.7368421052632	46.3157894736842	42.105263157894704
BDMS	60.0	54.2105263157895	54.2105263157895	37.368421052631604	MTLS	57.142857142857096	53.0612244897959	53.0612244897959	44.8979591836735
BEC	45.26315789473 68	53.1578947368421	48.421052631578895	48.421052631578895	PS	47.8947368421053	41.5789473684211	39.4736842105263	36.3157894736842
BH	50.0	50.0	47.8947368421053	49.4736842105263	PTT	43.684210526315795	58.9473684210526	58.9473684210526	58.9473684210526
BLA	46.842105263157904	52.631578947368396	52.631578947368396	45.7894736842105	PTTEP	46.842105263157904	53.1578947368421	52.1052631578947	54.2105263157895
BTS	46.842105263157904	55.789473684210 5	55.7894736842105	40.5263157894737	PTTGC	46.3157894736842	53.6842105263158	54.73684210526319	46.842105263157904
CBG	68.0	36.0	36.0	32.0	ROBINS	53.1578947368421	55.7894736842105	55.7894736842105	37.894736842105296
CENTEL	57.368421052631604	51.052631578947405	51.052631578947405	41.5789473684211	SAWAD	42.2222222222222	43.3333333333333	43.3333333333333	41.1111111111111
CK	54.2105263157895	46.842105263157904	41.5789473684211	38.9473684210526	SCB	44.7368421052632	55.7894736842105	55.7894736842105	41.0526315789474
CPALL	48.9473684210526	55.7894736842105	55.7894736842105	43.1578947368421	SCC	46.3157894736842	46.842105263157904	45.7894736842105	40.0
CPF	47.3684210526316	53.6842105263158	51.052631578947405	43.684210526315795	TASCO	53.6842105263158	44.7368421052632	43.684210526315795	46.3157894736842
CPN	55.26315789473 68	51.578947368421105	51.578947368421105	38.4210526315789	TCAP	55.7894736842105	51.052631578947405	51.052631578947405	38.4210526315789
DELTA	55.7894736842105	48.9473684210526	48.9473684210526	45.2631578947368	TMB	49.4736842105263	54.2105263157895	55.7894736842105	43.684210526315795
DTAC	50.0	48.9473684210526	46.842105263157904	50.0	TOP	60.0	44.7368421052632	44.7368421052632	40.0
EGCO	56.842105263157904	48.421052631578895	48.9473684210526	38.9473684210526	TPIPL	48.421052631578895	44.2105263157895	41.5789473684211	44.2105263157895
GLOW	53.1578947368421	44.7368421052632	43.1578947368421	41.5789473684211	TRUE	46.842105263157904	53.1578947368421	45.7894736842105	45.26315789473 68
GPSC	0.0	0.0	0.0	0.0	TTW	40.5263157894737	45.2631578947368	43.1578947368421	34.7368421052632
HMPRO	57.368421052631604	42.631578947368396	46.3157894736842	36.8421052631579	TU	52.1052631578947	55.2631578947368	55.2631578947368	40.0
INTUCH	42.105263157894704	43.1578947368421	43.1578947368421	42.105263157894704	WHA	55.2631578947368	52.1052631578947	50.0	44.7368421052632
IRPC	54.73684210526319	48.421052631578895	51.578947368421105	43.1578947368421	-	-	-	-	-

Table 5.41 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. It gives a pretty good result. But it has the same issue as the price prediction case. This MACD based model required quite a lot of data for training, so there is one stock, i.e., GPSC, that does not have enough records to use with this model. So these results are the average results over 48 stocks instead of 49 stocks from SET50.

Table 5.41 Average error for multi-class based model.

Base	MAPE
Close	50.8933676784554
Change	48.9253739507499
Tick	48.3596656986381
Direction	42.2928302900107

5.2.6 Revised Technical Analysis Prediction Model

Next, we do preprocessing indicator, i.e., EMA, and then use it to build the model.

$$ema10Change_n = ema10_n - ema10_{n-1}$$

$$ema60Change_n = ema60_n - ema60_{n-1}$$

$$ema250Change_n = ema250_n - ema250_{n-1}$$

$$close_n = f_{close}(close_{n-1}, ema10Change_{n-1}, ema60Change_{n-1}, ema250Change_{n-1})$$

$$change_n = f_{change}(change_{n-1}, ema10Change_{n-1}, ema60Change_{n-1}, ema250Change_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, ema10Change_{n-1}, ema60Change_{n-1}, ema250Change_{n-1})$$

$$direction_n = f_{direction}(direction_{n-1}, ema10Change_{n-1}, ema60Change_{n-1}, ema250Change_{n-1})$$

Table 5.42 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.42 MAPE results for EMA change based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	43.684210526315795	46.31578947368421	44.21052631578947	41.57894736842105	IVL	54.21052631578947	49.47368421052631	50.52631578947368	43.15789473684211
AOT	60.0	44.73684210526316	47.36842105263158	45.78947368421053	KBANK	44.21052631578947	51.57894736842105	46.31578947368421	47.36842105263158
BA	50.0	38.88888888888889	38.88888888888889	38.88888888888889	KCE	60.0	51.57894736842105	52.10526315789474	45.26315789473684
BANPU	41.57894736842105	46.842105263157904	47.36842105263158	43.15789473684211	KTB	41.57894736842105	48.421052631578945	46.842105263157904	39.47368421052631
BBL	44.73684210526316	42.105263157894704	42.105263157894704	41.57894736842105	LH	41.05263157894737	45.78947368421053	47.36842105263158	35.26315789473684
BCP	57.894736842105296	49.47368421052631	48.421052631578945	42.105263157894704	MINT	58.94736842105263	54.21052631578947	53.68421052631579	41.05263157894737
BDMS	60.52631578947369	45.78947368421053	45.26315789473684	39.47368421052631	MTLS	53.06122448979592	46.93877551020408	46.93877551020408	51.020408163265294
BEC	44.73684210526316	48.94736842105263	47.36842105263158	45.26315789473684	PS	47.36842105263158	44.21052631578947	42.105263157894704	36.31578947368421
BH	50.52631578947368	45.78947368421053	47.36842105263158	49.47368421052631	PTT	42.631578947368396	51.57894736842105	51.578947368421105	42.105263157894704
BLA	49.47368421052631	47.89473684210526	48.421052631578945	42.631578947368396	PTTEP	47.36842105263158	52.631578947368396	51.578947368421105	55.78947368421053
BTS	46.842105263157904	40.52631578947368	41.57894736842105	39.47368421052631	PTTGC	46.31578947368421	50.0	50.0	51.052631578947405
CBG	68.0	57.99999999999999	70.0	42.0	ROBINS	48.421052631578945	57.368421052631604	57.368421052631604	50.0
CENTEL	58.421052631578995	47.36842105263158	49.47368421052631	38.42105263157895	SAWAD	52.22222222222222	47.77777777777778	47.77777777777778	47.77777777777778
CK	61.052631578947405	47.36842105263158	47.36842105263158	38.94736842105263	SCB	41.57894736842105	51.578947368421105	51.052631578947405	45.78947368421053
CPALL	54.21052631578947	48.421052631578945	48.421052631578945	42.631578947368396	SCC	50.0	40.0	40.0	38.42105263157895
CPF	48.421052631578995	55.78947368421053	53.15789473684211	49.47368421052631	TASCO	53.68421052631579	47.36842105263158	50.0	50.0
CPN	55.26315789473684	51.052631578947405	51.052631578947405	50.52631578947368	TCAP	52.10526315789474	46.842105263157904	45.26315789473684	37.368421052631604
DELTA	51.578947368421105	45.26315789473684	44.21052631578947	45.26315789473684	TMB	52.631578947368396	49.47368421052631	48.94736842105263	43.684210526315795
DTAC	50.0	51.052631578947405	52.631578947368396	52.631578947368396	TOP	60.52631578947369	51.052631578947405	50.0	42.631578947368396
EGCO	58.421052631578995	44.73684210526316	43.684210526315795	41.05263157894737	TPIPL	54.21052631578947	42.631578947368396	43.15789473684211	44.736842105263158
GLOW	45.78947368421053	45.78947368421053	46.31578947368421	40.0	TRUE	54.73684210526316	46.842105263157904	44.21052631578947	41.57894736842105
GPSC	56.25	37.5	31.25	43.75	TTW	59.47368421052631	37.894736842105296	38.42105263157895	34.736842105263158
HMPRO	52.631578947368396	55.78947368421053	55.26315789473684	46.31578947368421	TU	46.31578947368421	47.89473684210526	46.31578947368421	46.31578947368421
INTUCH	43.684210526315795	41.57894736842105	42.105263157894704	42.105263157894704	WHA	59.47368421052631	46.31578947368421	45.78947368421053	47.89473684210526
IRPC	54.73684210526316	56.31578947368421	55.78947368421053	41.05263157894737	-	-	-	-	-

Table 5.43 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. It shows improvement for closing price and closing

change model over using EMA data directly, but gives lower accuracy for tick change and direction model.

Table 5.43 Average error for EMA change based model.

Base	MAPE
Close	51.6446138426728
Change	47.8120337286365
Tick	47.6823344805162
Direction	43.722131494918

Next, we use several data from technical analysis combine together to create a prediction model.

$$ema10Change_n = ema10_n - ema10_{n-1}$$

$$ema60Change_n = ema60_n - ema60_{n-1}$$

$$ema250Change_n = ema250_n - ema250_{n-1}$$

$$close_n = f_{close}(close_{n-1}, ema10Change_{n-1}, ema60Change_{n-1},$$
$$ema250Change_{n-1}, vol_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$change_n = f_{change}(change_{n-1}, ema10Change_{n-1}, ema60Change_{n-1},$$
$$ema250Change_{n-1}, vol_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$tick_n = f_{tick}(tick_{n-1}, ema10Change_{n-1}, ema60Change_{n-1},$$
$$ema250Change_{n-1}, vol_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

$$direction_n = f_{direction}(direction_{n-1}, ema10Change_{n-1}, ema60Change_{n-1},$$
$$ema250Change_{n-1}, vol_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

Table 5.44 shows the MAPE results of this model with EPS-BSVR and Laplace RBF kernel.

Table 5.45 shows the average error of EPS-BSVR with Laplace RBF kernel from all stocks by based data. The direction model gives the best accuracy so far,

Table 5.44 MAPE results for combine indicators based model.

Stock	Close	Change	Tick	Direction	Stock	Close	Change	Tick	Direction
ADVANC	42.631578947368396	44.2105263157895	46.842105263157904	43.684210526315795	IVL	54.2105263157895	55.2631578947368	53.6842105263158	38.9473684210526
AOT	60.52631578947369	52.10526315789474	52.631578947368396	35.26315789473684	KBANK	46.842105263157904	49.4736842105263	48.9473684210526	43.1578947368421
BA	61.11111111111111	38.88888888888889	38.88888888888889	38.88888888888889	KCE	59.4736842105263	50.0	50.0	40.0
BANPU	40.0	46.3157894736842	45.7894736842105	41.57894736842211	KTB	43.684210526315795	48.9473684210526	51.052631578947405	39.4736842105263
BBL	43.1578947368421	44.2105263157895	42.631578947368396	42.105263157894704	LH	38.9473684210526	50.0	48.421052631578895	38.4210526315789
BCP	57.894736842105296	36.8421052631579	37.368421052631604	38.9473684210526	MINT	59.4736842105263	42.105263157894704	42.631578947368396	38.9473684210526
BDMS	61.57894736842211	46.842105263157904	50.5263157894737	37.368421052631604	MTLS	57.142857142857096	44.8979591836735	44.8979591836735	48.9795918367347
BEC	45.7894736842105	52.1052631578947	52.631578947368396	43.1578947368421	PS	53.1578947368421	45.2631578947368	44.7368421052632	36.3157894736842
BH	50.5263157894737	56.3157894736842	51.578947368421105	49.4736842105263	PTT	41.5789473684211	49.4736842105263	48.421052631578895	44.7368421052632
BLA	48.9473684210526	52.631578947368396	53.1578947368421	45.7894736842105	PTTEP	47.3684210526316	53.1578947368421	52.631578947368396	54.73684210526319
BTS	51.578947368421105	46.31578947368421	43.684210526315795	41.0526315789474	PTTGC	51.578947368421105	53.1578947368421	54.2105263157895	50.0
CBG	68.0	68.0	68.0	32.0	ROBINS	46.3157894736842	55.7894736842105	56.842105263157904	43.684210526315795
CENTEL	57.894736842105296	43.1578947368421	44.7368421052632	38.4210526315789	SAWAD	52.2222222222222	48.88888888888889	47.7777777777778	41.1111111111111
CK	59.4736842105263	45.7894736842105	48.421052631578895	38.9473684210526	SCB	41.5789473684211	49.4736842105263	50.0	43.684210526315795
CPALL	54.73684210526319	55.2631578947368	55.2631578947368	43.1578947368421	SCC	52.631578947368396	45.2631578947368	45.2631578947368	41.57894736842211
CPF	45.7894736842105	48.421052631578895	50.0	43.684210526315795	TASCO	53.68421052631578	47.3684210526316	48.421052631578895	46.3157894736842
CPN	58.9473684210526	55.7894736842105	55.7894736842105	46.3157894736842	TCAP	54.73684210526319	52.631578947368396	52.631578947368396	38.4210526315789
DELTA	57.368421052631604	44.7368421052632	44.7368421052632	45.2631578947368	TMB	51.578947368421105	41.5789473684211	41.5789473684211	43.684210526315795
DTAC	50.5263157894737	46.3157894736842	44.2105263157895	51.052631578947405	TOP	59.4736842105263	46.3157894736842	46.842105263157904	42.631578947368396
EGCO	58.421052631578995	45.7894736842105	45.7894736842105	40.0	TPIPL	57.368421052631604	45.7894736842105	48.9473684210526	43.1578947368421
GLOW	53.1578947368421	43.684210526315795	44.2105263157895	38.9473684210526	TRUE	51.578947368421105	42.105263157894704	45.7894736842105	44.2105263157895
GPSC	56.25	56.25	56.25	43.75	TTW	57.368421052631604	40.0	38.9473684210526	34.7368421052632
HMPRO	50.0	56.842105263157904	57.368421052631604	36.842105263157	TU	51.052631578947405	43.1578947368421	44.7368421052632	38.4210526315789
INTUCH	40.52631578947737	43.684210526315795	44.7368421052632	42.105263157894704	WHA	58.9473684210526	44.2105263157895	44.7368421052632	37.368421052631604
IRPC	57.368421052631604	50.5263157894737	52.1052631578947	42.105263157894704	-	-	-	-	-

about 58.1%.

Table 5.45 Average error for combine indicators based model.

Base	MAPE
Close	52.5346913201371
Change	48.2723834610822
Tick	48.5612007423808
Direction	41.8902924220171

From all the results, we can improve the direction prediction accuracy by including additional derived data from technical indicator, and trading data. Next, we are going to build a combined model based on direction and price prediction model.

5.3 Combined Prediction Model

In this section, we are going to build a combined model based on direction and price prediction models in the previous sections. We use two models, i.e., one model for direction prediction, and one for price prediction. For the price prediction model, instead of predicting the actual value directly, we use two models, i.e., plus-model and minus-model. Plus-model is used to predict closing change and tick change in the plus area; while minus-model is used to predict value in the

minus area, i.e., predicted plus and minus lines in figure 5.10. Basically, we use the direction model to decide whether to use price prediction of the plus line, or the minus line.

For direction prediction, we use the combined indicators based model

$$direction_n = f_{direction}(direction_{n-1}, ema10Change_{n-1}, ema60Change_{n-1},$$
$$ema250Change_{n-1}, vol_{n-1}, buyVol_{n-1}, sellVol_{n-1})$$

For price prediction, we use the volume based model

$$change_n = f_{change}(change_{n-1}, vol_{n-1})$$
$$tick_n = f_{tick}(tick_{n-1}, vol_{n-1})$$

Table 5.46 shows the MAPE results of this model with EPS-BSVR and Bessel kernel for price prediction, and EPS-BSVR and Laplace RBF for direction prediction.

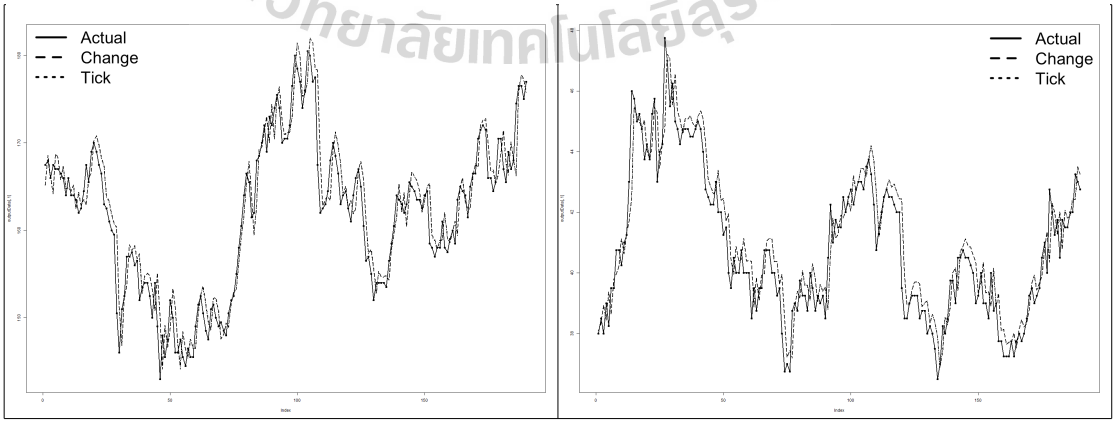

Figure 5.11 Graph of predicted results from two stocks for combined model.

Table 5.47 shows the average error of EPS-BSVR and Bessel kernel for

Table 5.46 MAPE results for combined model.

Stock	Change	Tick	Stock	Change	Tick
ADVANC	2.1373998382081	1.9357385059966499	IVL	2.22176462421284	2.1695316875523196
AOT	1.0944350550248	1.21077145533337	KBANK	1.54637370900912	1.48155051709721
BA	1.23157985268965	1.0703618403589301	KCE	1.5845477994407002	1.61588249166198
BANPU	2.31172684621091	2.3127986191378103	KTB	1.30761869526741	1.30761869526741
BBL	1.3139225795025	1.3012633680110999	LH	1.48378672142322	1.38542219715082
BCP	1.53985720877327	1.53985720877326	MINT	1.6550409127145602	1.75930849728278
BDMS	1.01815419740735	1.01815419740735	MTLS	1.02316599293839	1.02316599293839
BEC	2.2334356430252797	2.0407763968910397	PS	1.6123518465864701	1.94358465170683
BH	1.6163443986156898	1.82628316601148	PTT	1.9171966078992302	1.9253169208166099
BLA	1.8879090590837502	1.88790905908377	PTTEP	2.85059850679402	2.47861545905937
BTS	1.04886684522037	1.04118473036231	PTTGC	2.0293386381634497	2.02933863816344
CBG	1.9463380926644802	1.94633809266447	ROBINS	1.65102001794423	1.65101838553052
CENTEL	1.6768518384048001	1.67817158082363	SAWAD	1.68931658612193	1.68931658612194
CK	1.5697240270909099	1.63423753938805	SCB	1.6223596372827598	1.62244940682407
CPALL	1.4408256084995301	1.4408256084995301	SCC	1.2177093321346901	1.21938536101983
CPF	2.17285820376955	2.15747096524699	TASCO	2.8386199130929097	3.02978602108161
CPN	1.34770513249981	1.34770513249981	TCAP	1.21897137216341	1.21897190611956
DELTA	1.9926420003103	1.9926420003103	TMB	1.6422396333754201	1.64269086048019
DTAC	3.73552264565037	3.5689462964942202	TOP	1.60585765494137	1.60585765494137
EGCO	0.992513202414498	0.9946894193589351	TPIPL	2.05542999982555	2.2389309028069597
GLOW	1.4288650407232	1.43308929122464	TRUE	2.54587095559618	2.27700740066116
GPSC	1.70889299897349	1.70889299897348	TTW	0.9642102902514591	0.995124981613856
HMPRO	1.52655487932753	1.52305141641774	TU	1.4845906692541	1.4845906692541
INTUCH	1.77517042112579	1.77517042112579	WHA	1.8610123483480698	1.85943657849944
IRPC	1.4782106541506	1.56162163534382	-	-	-

price prediction, and EPS-BSVR and Laplace RBF for direction prediction from all stocks by based data.

Table 5.47 Average error for combined model.

Base	MAPE	MSE	R^2
Change	1.7113326272275	4.82374470079492	0.914527206844768
Tick	1.706160273661	4.97850801859148	0.912674695729356

Table 5.48 shows the average MAPE by correctness of direction from EPS-BSVR and Bessel kernel for price prediction, and EPS-BSVR and Laplace RBF for direction prediction from all stocks by based data.

Table 5.48 Average MAPE by direction correctness for combined model.

Base	MAPE (Correct Direction)	MAPE (Incorrect Direction)
Change	1.026682606194	2.6392502429388
Tick	1.0326148100817	2.6272351637939

Clearly from the result, MAPE results of the correct direction are better than those of the incorrect direction for both closing change, and tick change based

models. However, the overall error results are higher than those of volume based price prediction alone. This is not a surprise since our direction prediction model gives around 58% accuracy.

$$1.03 * 0.58 + 2.64 * 0.42 = 1.7062 \tag{5.1}$$

$$1.03 * 0.58 + 2.63 * 0.42 = 1.702 \tag{5.2}$$

In the next section, we are going to do the test with some existing prediction models for reference and comparison.

5.4 Reference Prediction Model

In this section, we use an existing prediction model to do an experiment with our dataset for reference and comparison. There are many people who believe that the stock market follows the random walk principle, thus it cannot be predicted. So "the best prediction for tomorrow's market price is simply today's price" (Yates, 2007). So we are going to use current value as predicted value for the next time period. We call it "same model".

ARIMA is a popular mathematical model for forecasting time series datasets. Neural network is also a popular tool in machine learning for forecasting.

For ARIMA, and NN, we use the forecast package of R for doing the experiment. Figure 5.12 shows charts of predicted results from these three prediction models. Table 5.49 shows average errors from these three models. The same model with closing price as based data gives the best result among them for price prediction. Table 5.50 shows directional error of these models. For direction prediction, the ARIMA model gives the best accuracy, about 54.6%, among these three models. And Table 5.51 shows average error by the correctness of direction.

For MAPE of the corrected direction, the same model of closing change gives the best results among them.

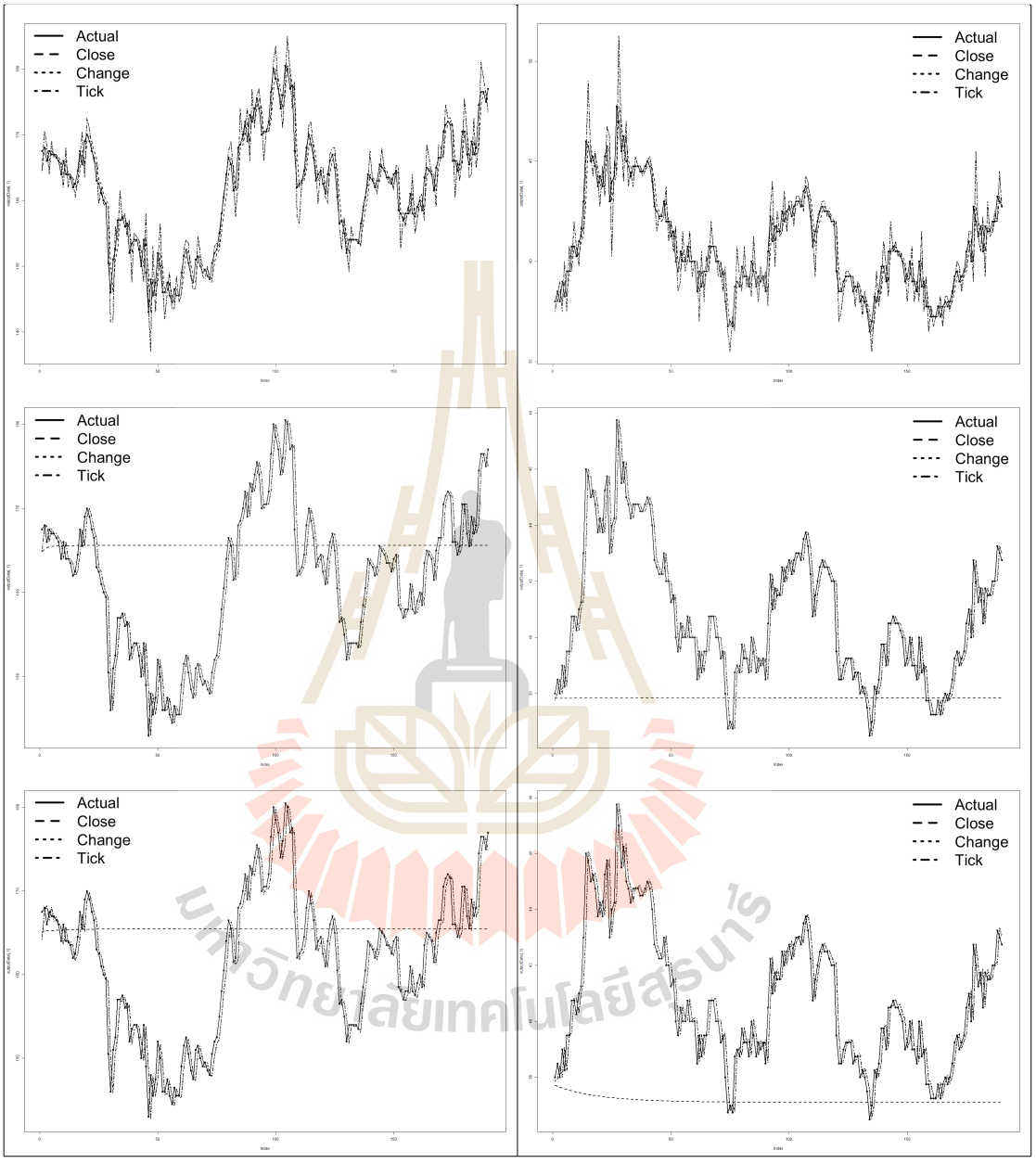

Figure 5.12 Graph of predicted results from two stocks.

Comparing the results from these three models to our prediction models. Table 5.52 shows the best average results of price prediction from three reference models, and an average result from our tick change with volume data model. It is clear that the same model from closing price data gives the best result. Our

Table 5.49 Average error.

Base	MAPE	MSE	R^2
Close (Same)	1.4707922791551	4.12081064061799	0.932063568421608
Change (Same)	2.1938384948462	8.2574309483378	0.86125545217694
Tick (Same)	2.203366741317	8.30588227337659	0.858920939549711
Close (ARIMA)	16.7340037954439	348.006148664653	-3.76166531842858
Change (ARIMA)	1.4855213786633	4.13162608431939	0.930609967710443
Tick (ARIMA)	1.4891451796515	4.13472609687953	0.930549217906697
Close (NN)	15.588071036827	374.90728434411	-3.2228709793623
Change (NN)	1.5024939866277	4.12989751362651	0.929397618350805
Tick (NN)	1.4944572315145	4.14062184597783	0.929705559381072

Table 5.50 Average direction error.

Base	MAPE
Same	51.8076678576909
ARIMA	45.3947532826235
NN	46.8501623348946

volume model from tick change gives slighly higher error than the ARIMA model from closing change.

Table 5.53 shows the best average results of direction prediction from three reference models, and an average result from our combined indicators model with direction data. Our model gives the best accuracry among these results.

Table 5.54 shows the best average results by correctness of direction from three reference models, and an average results from our combining model. When the predicted direction is correct, our model gives the best MAPE result.

Table 5.51 Average error by correctness of direction.

Base	MAPE (Correct Direction)	MAPE (Incorrect Direction)
Change (Same)	1.2931830100846	3.0329465310972
Tick (Same)	1.3096450463629	3.0361487812212
Change (ARIMA)	1.3349365805938	1.7036813101888
Tick (ARIMA)	1.3419476841018	1.7058569358793
Change (NN)	1.3242110884179	1.7485794844303
Tick (NN)	1.3589251970761	1.6934270996684

Table 5.52 Average error.

Base	MAPE	MSE	R^2
Tick (Volume)	1.4880198507158	4.15571623426123	0.931277232667798
Close (Same)	1.4707922791551	4.12081064061799	0.932063568421608
Change (ARIMA)	1.4855213786633	4.13162608431939	0.930609967710443
Tick (NN)	1.4944572315145	4.14062184597783	0.929705559381072

Table 5.53 Average direction error.

Base	MAPE
Direction (Combined Indicators)	41.8902924220171
Same	51.8076678576909
ARIMA	45.3947532826235
NN	46.8501623348946

Table 5.54 Average error by correctness of direction.

Base	MAPE (Correct Direction)	MAPE (Incorrect Direction)
Change (Combine)	1.026682606194	2.6392502429388
Change (Same)	1.2931830100846	3.0329465310972
Change (ARIMA)	1.3349365805938	1.7036813101888
Change (NN)	1.3242110884179	1.7485794844303

CHAPTER VI

CONCLUSION, DISCUSSION AND FUTURE WORK

In this study, we have introduced a new transformation for creating a new data set by using a certain characteristic of stock price movement, called tick change. With the rule of price movement for SET, the usual transformation, i.e., price difference, may not be stationary. From our test with our dataset from SET50, with a closing price, all stock fail the stationary test. With closing change, there are eight stocks that fails the test. With tick change, there are only two stocks that fail the test. So our transformation is more suitable to transform stock data from SET into stationary data.

We also use this transformed data set with a simple technical analysis calculation to create an indicator for buy-sell signal, i.e., SMA and EMA from closing change and tick change. With our dataset from SET50, our SMA from closing change and tick change give a better profit than using the popular MACD indicator.

For stock price prediction, our model based on closing change and tick change does better than the closing price model. Closing change and tick change give about the same results, just slightly higher or lower error. The tick change and volume based model gives the best result for our dataset from SET50. It does better than neural network model, but ARIMA with closing change and same model from closing price give better error.

For stock direction prediction, our combined indicators model from direc-

tion gives the best accuracy. It performs better than neural network, ARIMA, and same model.

By combining the price model and the direction model, we get the prediction model that give the best result when the forecasted direction is correct. But it gives poor result when its direction is incorrect. Thus the overall error is higher than the price model alone. So we need to improve the direction accuracy in order to decrease the overall error.

It is clear from our various prediction models that we can improve the performance of forecasting by including additional data, e.g., technical analysis indicator, preprocessed data from historical data of stock price. But using too many data may introduce additional error or noise into the model. The strategy to construct and select the suitable data is necessary.

Furthermore, we have only used historical data of stock to forecast a price of that particular stock in this study. We can improve our model by including information of the other relevant stocks, market index, or some kind of measurement for building a forecasting model. However, to achieve this, it would require analysis and strategies to identify which information is relevant and have impact on that particular stock. Obviously, each stock has its own set of relevant factors, so a mechanism to select the relevant factors is necessary.

REFERENCES

REFERENCES

Bassetti, E.M. (2007). **Technical Analysis of Stock Trends: Ninth Edition**. CRC Press.

Chaigusin, S., Chirathamjaree, C., and Clayden, J. (2008). Soft computing in the forecasting of the stock exchange of Thailand (SET). **Proceedings of IEEE Conference on Management of Innovation and Technology**. 1277-1281.

Chaigusin, S., Chirathamjaree, C., and Clayden, J. (2008). The Use of Neural Networks in the Prediction of the Stock Exchange of Thailand (SET) Index. **Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation**. 670-673.

Chen, W.H., and Shih, J.Y. (2006). Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. **International Journal of Electronic Finance**. 1(1): 49-67.

Fletcher, T. (2009). Support Vector Machines Explained. **Tutorial Paper**. University College London.

Gunn, S.R. (1998). Support Vector Machines for Classification and Regression. **Technical Report**. University of Southampton.

Huang, N.E., and Shen, S.S.P. (2005). **Hilbert-Huang Transform and Its Applications**. World Scientific, London.

Huang, W., Nakamori, Y., and Wang, S.Y. (2005). Forecasting stock market move-

ment direction with support vector machine. **Computers Operations Research**. 32(1): 2513-2522.

Iordanova, T. (2009). **Introduction To Stationary And Non-Stationary Processes.**

Kalyvasm, E. (2001). **Using Neural Networks and Genetic Algorithms to Predict Stock Market Returns**. Master Thesis, University of Manchester.

Kara, Y., Boyacioglu, M.A., and Baykan, O.K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. **Expert Systems with Applications**. 38(1): 5311-5319.

Karatzoglou, A., Meyer, D., and Hornik, K. (2006). Support vector machine in R. **Journal of Statistical Software**. 15(9): 1-28.

Kumar, M., and Thenmozhi, M. (2009). Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest. **Indian Institute of Capital Markets 9th Capital Markets Conference Paper**. 21(1): 1-16.

Leoywanichjalearn, Y. (2009) Factors Determinant Stock Market: New evidence of Thai stock market with Multiple Regression. **Asian Securities and Investments Federation**. eJournal Issue 12.

Premanode, B. (2013). **Prediction of Nonlinear Nonstationary Time Series Data Using a New Digital Filter and Support Vector Regression.** Ph.D. Thesis, Electrical & Electronic Engineering, Imperial College London.

Ray, S. (2012). Revisiting the Strength of Dow Theory in Assessing Stock Price

Movement. **Advances in Applied Economics and Finance**. 3(3): 591-598.

Rimcharoen, S., Sutivong, D., and Chongstitwatan, P. (2005). Prediction of the Stock Exchange of Thailand Using Adaptive Evolution Strategies. **Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence**. 232-236.

Smola, A.J., and Scholkopf, B. (2004). A tutorial on support vector regression. **Statistics and Computing**. 14(1): 199-222.

Sutheebanjard, P., and Premchaiswadi, W. (2010). Forecasting the Thailand Stock Market Using Evolution Strategies. **Asian Academy of Management Journal of Accounting and Finance**. 6(2): 85-114.

Sutheebanjard, P., and Premchaiswadi, W. (2010). Stock Exchange of Thailand Index prediction using Back Propagation Neural Networks. **2nd International Conference on Computer and Network Technology**. 377-380.

Tay, F.E.H., and Cao, L. (2001). Application of support vector machines in financial time series forecasting. **The International Journal of Management Science**. 29(1): 309-317.

The Stock Exchange of Thailand. (2016). **History of the Stock Exchange of Thailand**.

World Federation of Exchanges. **Monthly Reports**.

Yates, T. (2007). **4 Ways To Predict Market Performance.**

APPENDICES

APPENDIX A PROGRAMME FILES : R SCRIPTS

In this appendix, all R script programe used in this research is presented.

Price Prediction Script

Util : Utility Function

```
# define function to find a number of step for price
stepFromPrice <- function(price) {
    # range         step        #step       acc.#step
    # 0 - 2         0.01        200         0
    # 2 - 5         0.02        150         350
    # 5 - 10        0.05        100         450
    # 10 - 25       0.1         150         600
    # 25 - 100      0.25        300         900
    # 100 - 200     0.5         200         1100
    # 200 - 400     1           200         1300
    # 400 - ...     2           ...

    step <- 0
    if (price <= 0) {
        step <- 0
    } else if (price <= 2) {
        step <- price / 0.01
    } else if (price <= 5) {
        step <- 2 / 0.01 + (price - 2) / 0.02
    } else if (price <= 10) {
        step <- 2 / 0.01 + (5 - 2) / 0.02 + (price - 5) / 0.05
    } else if (price <= 25) {
        step <- 2 / 0.01 + (5 - 2) / 0.02 + (10 - 5) / 0.05 + (price - 10) / 0.1
    } else if (price <= 100) {
        step <- 2 / 0.01 + (5 - 2) / 0.02 + (10 - 5) / 0.05 + (25 - 10) / 0.1 + (price - 25) / 0.25
    } else if (price <= 200) {
        step <- 2 / 0.01 + (5 - 2) / 0.02 + (10 - 5) / 0.05 + (25 - 10) / 0.1 + (100 - 25) / 0.25 + (price - 100) /
            0.5
    } else if (price <= 400) {
        step <- 2 / 0.01 + (5 - 2) / 0.02 + (10 - 5) / 0.05 + (25 - 10) / 0.1 + (100 - 25) / 0.25 + (200 - 100) /
            0.5 + (price - 200) / 1
    } else {
        step <- 2 / 0.01 + (5 - 2) / 0.02 + (10 - 5) / 0.05 + (25 - 10) / 0.1 + (100 - 25) / 0.25 + (200 - 100) /
            0.5 + (400 - 200) / 1 + (price - 400) / 2
    }
    return(step)
}

# define function to find price from a number of step
priceFromStep <- function(step) {
    # range         step        #step       acc.#step
    # 0 - 2         0.01        200         200
    # 2 - 5         0.02        150         350
    # 5 - 10        0.05        100         450
    # 10 - 25       0.1         150         600
    # 25 - 100      0.25        300         900
    # 100 - 200     0.5         200         1100
    # 200 - 400     1           200         1300
    # 400 - ...     2           ...

    price <- 0
    if (step <= 0) {
        price <- 0
    } else if (step <= 200) {
        price <- 0.01 * step
    } else if (step <= 350) {
        price <- 0.01 * 200 + 0.02 * (step - 200)
    } else if (step <= 450) {
        price <- 0.01 * 200 + 0.02 * 150 + 0.05 * (step - 350)
    } else if (step <= 600) {
        price <- 0.01 * 200 + 0.02 * 150 + 0.05 * 100 + 0.1 * (step - 450)
    } else if (step <= 900) {
        price <- 0.01 * 200 + 0.02 * 150 + 0.05 * 100 + 0.1 * 150 + 0.25 * (step - 600)
    } else if (step <= 1100) {
        price <- 0.01 * 200 + 0.02 * 150 + 0.05 * 100 + 0.1 * 150 + 0.25 * 300 + 0.5 * (step - 900)
    } else if (step <= 1300) {
        price <- 0.01 * 200 + 0.02 * 150 + 0.05 * 100 + 0.1 * 150 + 0.25 * 300 + 0.5 * 200 + 1 * (step - 1100)
    } else {
        price <- 0.01 * 200 + 0.02 * 150 + 0.05 * 100 + 0.1 * 150 + 0.25 * 300 + 0.5 * 200 + 1 * 200 + 2 * (step -
            1300)
    }
    return(price)
}

# define function to find a tick size for price
tickSizeForPrice <- function(price) {
    # range         step        #step       acc.#step
    # 0 - 2         0.01        200         0
```

```
        # 2 - 5          0.02          150          350
76      # 5 - 10         0.05          100          450
        # 10 - 25        0.1           150          600
        # 25 - 100       0.25          300          900
        # 100 - 200      0.5           200          1100
        # 200 - 400      1             200          1300
81      # 400 - ...      2             ...

        tick <- 0
        if (price <= 0) {
            tick <- 0
86      } else if (price <= 2) {
            tick <- 0.01
        } else if (price <= 5) {
            tick <- 0.02
        } else if (price <= 10) {
91          tick <- 0.05
        } else if (price <= 25) {
            tick <- 0.1
        } else if (price <= 100) {
            tick <- 0.25
96      } else if (price <= 200) {
            tick <- 0.5
        } else if (price <= 400) {
            tick <- 1
        } else {
101         tick <- 2
        }
        return(tick)
    }

106 # define function for postprocessing predicted data (for price prediction from "Predict_Combine" script)
    # dataClose, dataChange, dataStepChange - full length of available data
    postProcessPredictionCombine <- function(dataClose, dataChange, dataStepChange, dataDirection, predictedDirection,
        predictedChange, predictedStepChange, numTest, numStep, outputDir) {
        ###### Calculate PredictedChange_Close, PredictedStepChange_Close ######
        # change
111     predictedChange_close <- rep(NA, numTest)
        for (i in 1:numTest) {
            predictedChange_close[i] <- dataClose[length(dataClose) - numTest + i - 1] + predictedChange[i]
        }
        # stepChange
116     predictedStepChange_close <- rep(NA, numTest)
        for (i in 1:numTest) {
            predictedStepChange_close[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
                + predictedStepChange[i])
        }

121     ###### Calculate Error ######
        close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
        close.sum2 <- rep(NA, numTest)
        for (i in 1:numTest) {
            close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
126     }
        close.tot <- sum(close.sum2, na.rm = TRUE)
        # change
        predictedChange_close.error <- rep(NA, numTest)
        predictedChange_close.error.mape <- vector()
131     predictedChange_close.error2 <- rep(NA, numTest)
        predictedChange_close.error2.mse <- vector()
        predictedChange_close.error2.r2 <- vector()
        if (numStep == 1) {
            for (i in 1:numTest) {
136             predictedChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - predictedChange_
                    close[i]) / dataClose[length(dataClose) - numTest + i]
                predictedChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - predictedChange_close[
                    i])^2
            }
            predictedChange_close.error.mape[1] <- mean(predictedChange_close.error, na.rm = TRUE)
            predictedChange_close.error2.mse[1] <- mean(predictedChange_close.error2, na.rm = TRUE)
141         predictedChange_close.error2.r2[1] <- 1 - sum(predictedChange_close.error2, na.rm = TRUE) / close.tot
        } else {
            for (i in 1:numTest) {
                predictedChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - predictedChange_
                    close[i, 1]) / dataClose[length(dataClose) - numTest + i]
                predictedChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - predictedChange_close[
                    i, 1])^2
146         }
            predictedChange_close.error.mape[1] <- mean(predictedChange_close.error, na.rm = TRUE)
            predictedChange_close.error2.mse[1] <- mean(predictedChange_close.error2, na.rm = TRUE)
            predictedChange_close.error2.r2[1] <- 1 - sum(predictedChange_close.error2, na.rm = TRUE) / close.tot
            for (i in 2:numStep) {
151             temp <- rep(NA, numTest)
                temp2 <- rep(NA, numTest)
                for (j in i:numTest) {
                    temp[j] <- abs(dataClose[length(dataClose) - numTest + j] - predictedChange_close[j, i]) /
                        dataClose[length(dataClose) - numTest + j]
                    temp2[j] <- (dataClose[length(dataClose) - numTest + j] - predictedChange_close[j, i])^2
156             }
                predictedChange_close.error <- cbind(predictedChange_close.error, temp)
                predictedChange_close.error.mape[i] <- mean(temp, na.rm = TRUE)
                predictedChange_close.error2 <- cbind(predictedChange_close.error2, temp2)
                predictedChange_close.error2.mse[i] <- mean(temp2, na.rm = TRUE)
161             predictedChange_close.error2.r2[i] <- 1 - sum(temp2, na.rm = TRUE) / close.tot
            }
```

```
        predictedChange_close.error.mape[numStep+1] <- mean(predictedChange_close.error, na.rm = TRUE)
        predictedChange_close.error2.mse[numStep+1] <- mean(predictedChange_close.error2, na.rm = TRUE)
        predictedChange_close.error2.r2[numStep+1] <- mean(predictedChange_close.error2.r2[1:numStep], na.rm = TRUE
            )
    }
    # stepChange
    predictedStepChange_close.error <- rep(NA, numTest)
    predictedStepChange_close.error.mape <- vector()
    predictedStepChange_close.error2 <- rep(NA, numTest)
    predictedStepChange_close.error2.mse <- vector()
    predictedStepChange_close.error2.r2 <- vector()
    if (numStep == 1) {
        for (i in 1:numTest) {
            predictedStepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] -
                predictedStepChange_close[i]) / dataClose[length(dataClose) - numTest + i]
            predictedStepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] -
                predictedStepChange_close[i])^2
        }
        predictedStepChange_close.error.mape[1] <- mean(predictedStepChange_close.error, na.rm = TRUE)
        predictedStepChange_close.error2.mse[1] <- mean(predictedStepChange_close.error2, na.rm = TRUE)
        predictedStepChange_close.error2.r2[1] <- 1 - sum(predictedStepChange_close.error2, na.rm = TRUE) / close.
            tot
    } else {
        for (i in 1:numTest) {
            predictedStepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] -
                predictedStepChange_close[i, 1]) / dataClose[length(dataClose) - numTest + i]
            predictedStepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] -
                predictedStepChange_close[i, 1])^2
        }
        predictedStepChange_close.error.mape[1] <- mean(predictedStepChange_close.error, na.rm = TRUE)
        predictedStepChange_close.error2.mse[1] <- mean(predictedStepChange_close.error2, na.rm = TRUE)
        predictedStepChange_close.error2.r2[1] <- 1 - sum(predictedStepChange_close.error2, na.rm = TRUE) / close.
            tot
        for (i in 2:numStep) {
            temp <- rep(NA, numTest)
            temp2 <- rep(NA, numTest)
            for (j in i:numTest) {
                temp[j] <- abs(dataClose[length(dataClose) - numTest + j] - predictedStepChange_close[j, i]) /
                    dataClose[length(dataClose) - numTest + j]
                temp2[j] <- (dataClose[length(dataClose) - numTest + j] - predictedStepChange_close[j, i])^2
            }
            predictedStepChange_close.error <- cbind(predictedStepChange_close.error, temp)
            predictedStepChange_close.error.mape[i] <- mean(temp, na.rm = TRUE)
            predictedStepChange_close.error2 <- cbind(predictedStepChange_close.error2, temp2)
            predictedStepChange_close.error2.mse[i] <- mean(temp2, na.rm = TRUE)
            predictedStepChange_close.error2.r2[i] <- 1 - sum(temp2, na.rm = TRUE) / close.tot
        }
        predictedStepChange_close.error.mape[numStep+1] <- mean(predictedStepChange_close.error, na.rm = TRUE)
        predictedStepChange_close.error2.mse[numStep+1] <- mean(predictedStepChange_close.error2, na.rm = TRUE)
        predictedStepChange_close.error2.r2[numStep+1] <- mean(predictedStepChange_close.error2.r2[1:numStep], na.
            rm = TRUE)
    }

    ###### Calculate Direction Error ######
    dataDirection.error <- rep(NA, numTest)
    dataDirection.error.mape <- vector()
    for (i in 1:numTest) {
        dataDirection.error[i] <- abs(dataDirection[length(dataDirection) - numTest + i] - predictedDirection[i]) /
            2
    }
    dataDirection.error.mape[1] <- mean(dataDirection.error, na.rm = TRUE)

    ###### Calculate Error by correctness of Direction ######
    predictedChange_close.direction_correct.error <- rep(NA, numTest)
    predictedChange_close.direction_correct.error.mape <- vector()
    predictedChange_close.direction_wrong.error <- rep(NA, numTest)
    predictedChange_close.direction_wrong.error.mape <- vector()
    predictedStepChange_close.direction_correct.error <- rep(NA, numTest)
    predictedStepChange_close.direction_correct.error.mape <- vector()
    predictedStepChange_close.direction_wrong.error <- rep(NA, numTest)
    predictedStepChange_close.direction_wrong.error.mape <- vector()
    for (i in 1:numTest) {
        if (dataDirection[length(dataDirection) - numTest + i] == predictedDirection[i]) {
            ## correct direction
            predictedChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] -
                predictedChange_close[i]) / dataClose[length(dataClose) - numTest + i]
            predictedStepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i]
                - predictedStepChange_close[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            predictedChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] -
                predictedChange_close[i]) / dataClose[length(dataClose) - numTest + i]
            predictedStepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] -
                predictedStepChange_close[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    predictedChange_close.direction_correct.error.mape[1] <- mean(predictedChange_close.direction_correct.error, na
        .rm = TRUE)
    predictedChange_close.direction_wrong.error.mape[1] <- mean(predictedChange_close.direction_wrong.error, na.rm
        = TRUE)
    predictedStepChange_close.direction_correct.error.mape[1] <- mean(predictedStepChange_close.direction_correct.
        error, na.rm = TRUE)
    predictedStepChange_close.direction_wrong.error.mape[1] <- mean(predictedStepChange_close.direction_wrong.error
        , na.rm = TRUE)
```

```
    ##### Prepare output data #####
241 outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)], dataStepChange[-1:-(length(dataStepChange) - numTest)], predictedChange, predictedStepChange, predictedChange_close, predictedStepChange_close, predictedChange_close.error, predictedStepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
246 colnames(outputData)[4] <- "PredictedChange_1"
    colnames(outputData)[5] <- "PredictedStepChange_1"
    colnames(outputData)[6] <- "PredictedChange_Close_1"
    colnames(outputData)[7] <- "PredictedStepChange_Close_1"
    colnames(outputData)[8] <- "PredictedChange_Close_Error_1"
251 colnames(outputData)[9] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(predictedChange_close.error.mape, predictedStepChange_close.error.mape, predictedChange_close.error2.mse, predictedStepChange_close.error2.mse, predictedChange_close.error2.r2, predictedStepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Change_MAPE"
256 colnames(errorData)[2] <- "StepChange_MAPE"
    colnames(errorData)[3] <- "Change_MSE"
    colnames(errorData)[4] <- "StepChange_MSE"
    colnames(errorData)[5] <- "Change_R2"
    colnames(errorData)[6] <- "StepChange_R2"
261
    errorByDirectionData <- cbind(dataDirection.error.mape, predictedChange_close.direction_correct.error.mape, predictedChange_close.direction_wrong.error.mape, predictedStepChange_close.direction_correct.error.mape, predictedStepChange_close.direction_wrong.error.mape)
    errorByDirectionData <- as.data.frame(errorByDirectionData)
    colnames(errorByDirectionData)[1] <- "Direction_MAPE"
    colnames(errorByDirectionData)[2] <- "Change_Correct_MAPE"
266 colnames(errorByDirectionData)[3] <- "Change_Wrong_MAPE"
    colnames(errorByDirectionData)[4] <- "StepChange_Correct_MAPE"
    colnames(errorByDirectionData)[5] <- "StepChange_Wrong_MAPE"

    ##### Write data to csv file #####
271 # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)
276
    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

    errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
281 write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

    ##### Plot Graph #####
    # find min, max value to define the y range of graph
    minValue <- min(outputData[,1], predictedChange_close, predictedStepChange_close)
286 maxValue <- max(outputData[,1], predictedChange_close, predictedStepChange_close)
    if (is.finite(minValue) == FALSE) {
        minValue = 0
    }
    if (is.finite(maxValue) == FALSE) {
291     maxValue = 400
    }
    for (i in 1:numStep) {
        outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
        # Start PNG device driver to save output to PNG
296     png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
        # Graph using blue points overlayed by a line
        plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
        if (numStep == 1) {
            lines(predictedChange_close, type = "l", col = "black", lty = 2, lwd = 2)
301         lines(predictedStepChange_close, type = "l", col = "black", lty = 3, lwd = 2)
        } else {
            lines(predictedChange_close[,i], type = "l", col = "black", lty = 2, lwd = 2)
            lines(predictedStepChange_close[,i], type = "l", col = "black", lty = 3, lwd = 2)
        }
306     # Create box around plot
        box()
        # Create a legend in the top-left corner that is slightly
        # smaller and has no border
        legend("topleft", legend = c("Close", "PredictedChangeClose", "PredictedStepChangeClose"), cex = 0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
311     # Turn off device driver (to flush output to png)
        dev.off()
    }
}
```

PredictPrice1 : Time Series

```
1  ##### ====== Begin ====== #####
   # clear screen
   rm(list=ls())
   # include function
   source("Util.R")
6
   ##### Read command line arguments #####
   # arguments: inputFileName outputDir svmType kernelType
```

```
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# pair of data: (index, current)
pairClose <- as.data.frame(cbind(1:length(dataClose), dataClose))
colnames(pairClose)[1] <- "Index"
colnames(pairClose)[2] <- "PredictedClose"
pairChange <- as.data.frame(cbind(1:length(dataChange), dataChange))
colnames(pairChange)[1] <- "Index"
colnames(pairChange)[2] <- "PredictedChange"
pairStepChange <- as.data.frame(cbind(1:length(dataStepChange), dataStepChange))
colnames(pairStepChange)[1] <- "Index"
colnames(pairStepChange)[2] <- "PredictedStepChange"

# split into 2 variable: train, test
pairClose.train <- as.data.frame(pairClose[1:(length(pairClose[,1]) - numTest),])
pairClose.test <- as.data.frame(pairClose[-1:-(length(pairClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
pairChange.train <- as.data.frame(pairChange[2:(length(pairChange[,1]) - numTest),])
pairChange.test <- as.data.frame(pairChange[-1:-(length(pairChange[,1]) - numTest),])
pairStepChange.train <- as.data.frame(pairStepChange[2:(length(pairStepChange[,1]) - numTest),])
pairStepChange.test <- as.data.frame(pairStepChange[-1:-(length(pairStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = pairClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = pairChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = pairStepChange.train, type = svmType, kernel =
    kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- predict(svr.close.model, pairClose.test)
# change
svr.change.predict <- predict(svr.change.model, pairChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, pairStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
```

```
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
```

```
            dataDirection.change.error[i] <- 0
        } else {
            dataDirection.change.error[i] <- 1
        }
        if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.stepChange.error[i] <- 0
        } else {
            dataDirection.stepChange.error[i] <- 1
        }
    }
    dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
    dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
    dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

    ##### Calculate Error by correctness of Direction #####
    svr.close.direction_correct.error <- rep(NA, numTest)
    svr.close.direction_correct.error.mape <- vector()
    svr.close.direction_wrong.error <- rep(NA, numTest)
    svr.close.direction_wrong.error.mape <- vector()
    svr.change_close.direction_correct.error <- rep(NA, numTest)
    svr.change_close.direction_correct.error.mape <- vector()
    svr.change_close.direction_wrong.error <- rep(NA, numTest)
    svr.change_close.direction_wrong.error.mape <- vector()
    svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
    svr.stepChange_close.direction_correct.error.mape <- vector()
    svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
    svr.stepChange_close.direction_wrong.error.mape <- vector()
    for (i in 1:numTest) {
        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
            ## correct direction
            svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
                i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
                ]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
            ## correct direction
            svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
            ## correct direction
            svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
    svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
    svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
    svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
    svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
        TRUE)
    svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
        )

    ##### Prepare output data #####
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
        predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
        stepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "PredictedClose_1"
    colnames(outputData)[5] <- "PredictedChange_1"
    colnames(outputData)[6] <- "PredictedStepChange_1"
    colnames(outputData)[7] <- "PredictedChange_Close_1"
    colnames(outputData)[8] <- "PredictedStepChange_Close_1"
    colnames(outputData)[9] <- "PredictedClose_Error_1"
    colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
    colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
        error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
        close.error2.r2, svr.stepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Close_MSE"
    colnames(errorData)[5] <- "Change_MSE"
    colnames(errorData)[6] <- "StepChange_MSE"
```

```
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()

##### ====== End ====== #####
```

PredictPrice2 : Historical Data

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
```

```
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (prev_1, prev_2, ..., prev_n, next)
# prev_i: remove first i-1 rows and last n-i+1 rows
# next: remove first n rows
setClose <- dataClose[-(numRecord-numUse):-numRecord]
setChange <- dataChange[-(numRecord-numUse):-numRecord]
setStepChange <- dataStepChange[-(numRecord-numUse):-numRecord]
for (i in 2:numUse)
{
    tempSetClose <- dataClose[-(numRecord-numUse+i-1):-numRecord]
    tempSetChange <- dataChange[-(numRecord-numUse+i-1):-numRecord]
    tempSetStepChange <- dataStepChange[-(numRecord-numUse+i-1):-numRecord]

    setClose <- cbind(setClose, tempSetClose[-1:-(i-1)])
    setChange <- cbind(setChange, tempSetChange[-1:-(i-1)])
    setStepChange <- cbind(setStepChange, tempSetStepChange[-1:-(i-1)])
}
setClose <- as.data.frame(cbind(setClose, dataClose[-1:-numUse]))
setChange <- as.data.frame(cbind(setChange, dataChange[-1:-numUse]))
setChange <- setChange[-1,]
setStepChange <- as.data.frame(cbind(setStepChange, dataStepChange[-1:-numUse]))
setStepChange <- setStepChange[-1,]
for (i in 1:numUse)
{
    colnames(setClose)[i] <- paste("Close", i, sep = "_")
    colnames(setChange)[i] <- paste("Change", i, sep = "_")
    colnames(setStepChange)[i] <- paste("StepChange", i, sep = "_")
}
colnames(setClose)[numUse+1] <- "PredictedClose"
colnames(setChange)[numUse+1] <- "PredictedChange"
colnames(setStepChange)[numUse+1] <- "PredictedStepChange"
# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- predict(svr.close.model, setClose.test)
# change
svr.change.predict <- predict(svr.change.model, setChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, setStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
```

```
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
```

```
            dataDirection.stepChange.error[i] <- 0
        } else {
            dataDirection.stepChange.error[i] <- 1
        }
    }
    dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
    dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
    dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

    ##### Calculate Error by correctness of Direction #####
    svr.close.direction_correct.error <- rep(NA, numTest)
    svr.close.direction_correct.error.mape <- vector()
    svr.close.direction_wrong.error <- rep(NA, numTest)
    svr.close.direction_wrong.error.mape <- vector()
    svr.change_close.direction_correct.error <- rep(NA, numTest)
    svr.change_close.direction_correct.error.mape <- vector()
    svr.change_close.direction_wrong.error <- rep(NA, numTest)
    svr.change_close.direction_wrong.error.mape <- vector()
    svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
    svr.stepChange_close.direction_correct.error.mape <- vector()
    svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
    svr.stepChange_close.direction_wrong.error.mape <- vector()
    for (i in 1:numTest) {
        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
            ## correct direction
            svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
                i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
                ]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
            ## correct direction
            svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
            ## correct direction
            svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
    svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
    svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
    svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
    svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
        TRUE)
    svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
        )

    ##### Prepare output data #####
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
        predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
        stepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "PredictedClose_1"
    colnames(outputData)[5] <- "PredictedChange_1"
    colnames(outputData)[6] <- "PredictedStepChange_1"
    colnames(outputData)[7] <- "PredictedChange_Close_1"
    colnames(outputData)[8] <- "PredictedStepChange_Close_1"
    colnames(outputData)[9] <- "PredictedClose_Error_1"
    colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
    colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
        error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
        close.error2.r2, svr.stepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Close_MSE"
    colnames(errorData)[5] <- "Change_MSE"
    colnames(errorData)[6] <- "StepChange_MSE"
    colnames(errorData)[7] <- "Close_R2"
    colnames(errorData)[8] <- "Change_R2"
    colnames(errorData)[9] <- "StepChange_R2"

    errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
```

```
        stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
        change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
        direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ===== End ===== #####
```

PredictPrice3 : EMA

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "$", sep = ""), colnames(inputData))
ema10ColumnIndex = grep("EMAV.C.10.", colnames(inputData))
ema60ColumnIndex = grep("EMAV.C.60.", colnames(inputData))
ema250ColumnIndex = grep("EMAV.C.250.", colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataEMA10 <- inputData[,ema10ColumnIndex]
dataEMA60 <- inputData[,ema60ColumnIndex]
dataEMA250 <- inputData[,ema250ColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
```

```
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, ema10, ema60, ema250, next)
# current, ema10, ema60, ema250: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord], dataEMA250[-
    numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "EMA10"
colnames(setClose)[3] <- "EMA60"
colnames(setClose)[4] <- "EMA250"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord], dataEMA250[-
    numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "EMA10"
colnames(setChange)[3] <- "EMA60"
colnames(setChange)[4] <- "EMA250"
colnames(setChange)[5] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord],
    dataEMA250[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "EMA10"
colnames(setStepChange)[3] <- "EMA60"
colnames(setStepChange)[4] <- "EMA250"
colnames(setStepChange)[5] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- predict(svr.close.model, setClose.test)
# change
svr.change.predict <- predict(svr.change.model, setChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, setStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
```

```
    for (i in 1:numTest) {
        close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
132 }
    close.tot <- sum(close.sum2, na.rm = TRUE)
    # close
    svr.close.error <- rep(NA, numTest)
    svr.close.error.mape <- vector()
137 svr.close.error2 <- rep(NA, numTest)
    svr.close.error2.mse <- vector()
    svr.close.error2.r2 <- vector()
    for (i in 1:numTest) {
        svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
            (dataClose) - numTest + i]
142     svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
    }
    svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
    svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
    svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
147 # change
    svr.change_close.error <- rep(NA, numTest)
    svr.change_close.error.mape <- vector()
    svr.change_close.error2 <- rep(NA, numTest)
    svr.change_close.error2.mse <- vector()
152 svr.change_close.error2.r2 <- vector()
    for (i in 1:numTest) {
        svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
            dataClose[length(dataClose) - numTest + i]
        svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
    }
157 svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
    svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
    svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
    # stepChange
    svr.stepChange_close.error <- rep(NA, numTest)
162 svr.stepChange_close.error.mape <- vector()
    svr.stepChange_close.error2 <- rep(NA, numTest)
    svr.stepChange_close.error2.mse <- vector()
    svr.stepChange_close.error2.r2 <- vector()
    for (i in 1:numTest) {
167     svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
        svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
            ])^2
    }
    svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
    svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
172 svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

    ##### Calculate Direction #####
    dataDirection.close <- rep(NA, numTest)
    dataDirection.change <- rep(NA, numTest)
177 dataDirection.stepChange <- rep(NA, numTest)
    for (i in 1:numTest) {
        if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
            dataDirection.close[i] <- -1
        } else {
182         dataDirection.close[i] <- 1
        }
        if (svr.change.predict[i] < 0) {
            dataDirection.change[i] <- -1
        } else {
187         dataDirection.change[i] <- 1
        }
        if (svr.stepChange.predict[i] < 0) {
            dataDirection.stepChange[i] <- -1
        } else {
192         dataDirection.stepChange[i] <- 1
        }
    }

    ##### Calculate Direction Error #####
197 dataDirection.close.error <- rep(NA, numTest)
    dataDirection.close.error.mape <- vector()
    dataDirection.change.error <- rep(NA, numTest)
    dataDirection.change.error.mape <- vector()
    dataDirection.stepChange.error <- rep(NA, numTest)
202 dataDirection.stepChange.error.mape <- vector()
    for (i in 1:numTest) {
        if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.close.error[i] <- 0
        } else {
207         dataDirection.close.error[i] <- 1
        }
        if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.change.error[i] <- 0
        } else {
212         dataDirection.change.error[i] <- 1
        }
        if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.stepChange.error[i] <- 0
        } else {
217         dataDirection.stepChange.error[i] <- 1
        }
    }
```

```
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
            ]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
```

```
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

###### Write data to csv file ######
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

###### Plot Graph ######
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice4 : Volume

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
volColumnIndex = grep(paste("^", "Vol", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataVol <- inputData[,volColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
```

```
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

###### Prepare Train Data ######
# set of data close: (current_close, vol, next_close)
# current_close, current_vol, close, vol: remove last row
# next_close: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataVol[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "Vol"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataVol[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "Vol"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataVol[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "Vol"
colnames(setStepChange)[3] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

###### Train SVM ######
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

###### Predict 1 ahead ######
# close
svr.close.predict <- predict(svr.close.model, setClose.test)
# change
svr.change.predict <- predict(svr.change.model, setChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, setStepChange.test)

###### Calculate PredictedChange_Close, PredictedStepChange_Close ######
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

###### Calculate Error ######
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
```

```
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

###### Calculate Direction ######
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

###### Calculate Direction Error ######
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

###### Calculate Error by correctness of Direction ######
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
```

```
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
            ]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
```

```
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ====== End ====== #####
```

PredictPrice5 : Buy-Sell Volume

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
buyVolColumnIndex = grep(paste("Buy.Vol", "\$", sep = ""), colnames(inputData))
sellVolColumnIndex = grep(paste("Sell.Vol", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataBuyVol <- inputData[,buyVolColumnIndex]
dataSellVol <- inputData[,sellVolColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
```

```
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data close: (current_close, buyVol, sellVol, next_close)
# current_close, current_buyVol, current_sellVol, close, buyVol, sellVol: remove last row
# next_close: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord], dataClose
    [-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "BuyVol"
colnames(setClose)[3] <- "SellVol"
colnames(setClose)[4] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "BuyVol"
colnames(setChange)[3] <- "SellVol"
colnames(setChange)[4] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "BuyVol"
colnames(setStepChange)[3] <- "SellVol"
colnames(setStepChange)[4] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- predict(svr.close.model, setClose.test)
# change
svr.change.predict <- predict(svr.change.model, setChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, setStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
```

```
            dataClose[length(dataClose) - numTest + i]
        svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
    }
    svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
    svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
    svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
    # stepChange
    svr.stepChange_close.error <- rep(NA, numTest)
    svr.stepChange_close.error.mape <- vector()
    svr.stepChange_close.error2 <- rep(NA, numTest)
    svr.stepChange_close.error2.mse <- vector()
    svr.stepChange_close.error2.r2 <- vector()
    for (i in 1:numTest) {
        svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
        svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
            ])^2
    }
    svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
    svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
    svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

    ##### Calculate Direction #####
    dataDirection.close <- rep(NA, numTest)
    dataDirection.change <- rep(NA, numTest)
    dataDirection.stepChange <- rep(NA, numTest)
    for (i in 1:numTest) {
        if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
            dataDirection.close[i] <- -1
        } else {
            dataDirection.close[i] <- 1
        }
        if (svr.change.predict[i] < 0) {
            dataDirection.change[i] <- -1
        } else {
            dataDirection.change[i] <- 1
        }
        if (svr.stepChange.predict[i] < 0) {
            dataDirection.stepChange[i] <- -1
        } else {
            dataDirection.stepChange[i] <- 1
        }
    }

    ##### Calculate Direction Error #####
    dataDirection.close.error <- rep(NA, numTest)
    dataDirection.close.error.mape <- vector()
    dataDirection.change.error <- rep(NA, numTest)
    dataDirection.change.error.mape <- vector()
    dataDirection.stepChange.error <- rep(NA, numTest)
    dataDirection.stepChange.error.mape <- vector()
    for (i in 1:numTest) {
        if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.close.error[i] <- 0
        } else {
            dataDirection.close.error[i] <- 1
        }
        if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.change.error[i] <- 0
        } else {
            dataDirection.change.error[i] <- 1
        }
        if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.stepChange.error[i] <- 0
        } else {
            dataDirection.stepChange.error[i] <- 1
        }
    }
    dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
    dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
    dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

    ##### Calculate Error by correctness of Direction #####
    svr.close.direction_correct.error <- rep(NA, numTest)
    svr.close.direction_correct.error.mape <- vector()
    svr.close.direction_wrong.error <- rep(NA, numTest)
    svr.close.direction_wrong.error.mape <- vector()
    svr.change_close.direction_correct.error <- rep(NA, numTest)
    svr.change_close.direction_correct.error.mape <- vector()
    svr.change_close.direction_wrong.error <- rep(NA, numTest)
    svr.change_close.direction_wrong.error.mape <- vector()
    svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
    svr.stepChange_close.direction_correct.error.mape <- vector()
    svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
    svr.stepChange_close.direction_wrong.error.mape <- vector()
    for (i in 1:numTest) {
        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
            ## correct direction
            svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
                i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
                ]) / dataClose[length(dataClose) - numTest + i]
```

```
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
```

```
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice6 : Buy-Sell Volume with ATO/ATC

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
buyVolColumnIndex = grep(paste("Buy.Vol", "\$", sep = ""), colnames(inputData))
sellVolColumnIndex = grep(paste("Sell.Vol", "\$", sep = ""), colnames(inputData))
atoatcColumnIndex = grep(paste("ATO.ATC", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataBuyVol <- inputData[,buyVolColumnIndex]
dataSellVol <- inputData[,sellVolColumnIndex]
dataATOATC <- inputData[,atoatcColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data close: (current_close, buyVol, sellVol, atoatc, next_close)
# current_close, current_buyVol, current_sellVol, current_atoatc, close, buyVol, sellVol, atoatc: remove last row
```

```
# next_close: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord], dataATOATC
    [-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "BuyVol"
colnames(setClose)[3] <- "SellVol"
colnames(setClose)[4] <- "ATOATC"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataATOATC[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "BuyVol"
colnames(setChange)[3] <- "SellVol"
colnames(setChange)[4] <- "ATOATC"
colnames(setChange)[5] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataATOATC[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "BuyVol"
colnames(setStepChange)[3] <- "SellVol"
colnames(setStepChange)[4] <- "ATOATC"
colnames(setStepChange)[5] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])

# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])

setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- predict(svr.close.model, setClose.test)
# change
svr.change.predict <- predict(svr.change.model, setChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, setStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
```

```
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
            ]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
```

```
            ## correct direction
            svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
            ## correct direction
            svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
    svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
    svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
    svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
    svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
        TRUE)
    svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
        )

    ##### Prepare output data #####
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
        predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
        stepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "PredictedClose_1"
    colnames(outputData)[5] <- "PredictedChange_1"
    colnames(outputData)[6] <- "PredictedStepChange_1"
    colnames(outputData)[7] <- "PredictedChange_Close_1"
    colnames(outputData)[8] <- "PredictedStepChange_Close_1"
    colnames(outputData)[9] <- "PredictedClose_Error_1"
    colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
    colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
        error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
        close.error2.r2, svr.stepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Close_MSE"
    colnames(errorData)[5] <- "Change_MSE"
    colnames(errorData)[6] <- "StepChange_MSE"
    colnames(errorData)[7] <- "Close_R2"
    colnames(errorData)[8] <- "Change_R2"
    colnames(errorData)[9] <- "StepChange_R2"

    errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
        stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
        change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
        direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
    errorByDirectionData <- as.data.frame(errorByDirectionData)
    colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
    colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
    colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
    colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
    colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
    colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
    colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
    colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
    colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

    errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
    write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

    ##### Plot Graph #####
    # find min, max value to define the y range of graph
    minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
    maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
    if (is.finite(minValue) == FALSE) {
        minValue = 0
    }
```

```
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice7 : CMF (Chaikin Money Flow)

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

###### Read command line arguments ######
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

###### Prepare Input Data ######
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
cmfColumnIndex = grep(paste("^", "CMF", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataCMF <- inputData[,cmfColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

###### Prepare Train Data ######
# set of data: (current, cmf, next)
# current, cmf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataCMF[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "CMF"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataCMF[-numRecord], dataChange[-1]))
```

```
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "CMF"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataCMF[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "CMF"
colnames(setStepChange)[3] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
```

```
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

###### Calculate Direction ######
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

###### Calculate Direction Error ######
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

###### Calculate Error by correctness of Direction ######
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
```

```
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
```

```
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice8 : MOF (Money Flow)

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments ######
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data ######
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
mofColumnIndex = grep(paste("^", "MOF", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataMOF <- inputData[,mofColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data ######
# set of data: (current, mof, next)
# current, mof: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataMOF[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "MOF"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataMOF[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "MOF"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataMOF[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "MOF"
colnames(setStepChange)[3] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
```

```
##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
```

```
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
```

```
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ====== End ====== #####
```

PredictPrice9 : OBV (On Balance Volume)

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
```

```
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
obvColumnIndex = grep(paste("^", "OBV", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataOBV <- inputData[,obvColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, obv, next)
# current, obv: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataOBV[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "OBV"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataOBV[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "OBV"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataOBV[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "OBV"
colnames(setStepChange)[3] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
```

```
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
```

```
            dataDirection.close.error[i] <- 0
        } else {
            dataDirection.close.error[i] <- 1
        }
        if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.change.error[i] <- 0
        } else {
            dataDirection.change.error[i] <- 1
        }
        if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
            dataDirection.stepChange.error[i] <- 0
        } else {
            dataDirection.stepChange.error[i] <- 1
        }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
```

```
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ===== End ===== #####
```

PredictPrice10 : RSI (Relative Strength Index)

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
rsiColumnIndex = grep(paste("^", "RSI.14.", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataRSI <- inputData[,rsiColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
```

```
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, rsi, next)
# current, rsi: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataRSI[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "RSI"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataRSI[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "RSI"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataRSI[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "RSI"
colnames(setStepChange)[3] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
```

```
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
```

```
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
```

```
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ===== End ===== #####
```

PredictPrice11 : ADX (Average Directional Index), DIP (Directional Index Plus), and DIM (Directional Index Minus)

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
adxColumnIndex = grep(paste("^", "ADX", "\$", sep = ""), colnames(inputData))
dipColumnIndex = grep(paste("^", "DI.", "\$", sep = ""), colnames(inputData))
dimColumnIndex = grep(paste("^", "DI..1", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataADX <- inputData[,adxColumnIndex]
dataDIP <- inputData[,dipColumnIndex]
dataDIM <- inputData[,dimColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
```

```
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, adx, dip, dim, next)
# current, adx, dip, dim: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM[-numRecord
    ], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "ATR"
colnames(setClose)[3] <- "DIP"
colnames(setClose)[4] <- "DIM"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM[-
    numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "ATR"
colnames(setChange)[3] <- "DIP"
colnames(setChange)[4] <- "DIM"
colnames(setChange)[5] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM
    [-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "ATR"
colnames(setStepChange)[3] <- "DIP"
colnames(setStepChange)[4] <- "DIM"
colnames(setStepChange)[5] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
```

```
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

###### Calculate Direction ######
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

###### Calculate Direction Error ######
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
```

```
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
```

```
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ====== End ====== #####
```

PredictPrice12 : ATR (Average True Range)

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])


##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
atrColumnIndex = grep(paste("^", "ATR", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataATR <- inputData[,atrColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
```

```
        step <- stepFromPrice(dataClose[i])
        previousStep <- stepFromPrice(dataClose[i-1])
        dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, atr, next)
# current, atr: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataATR[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "ATR"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataATR[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "ATR"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataATR[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "ATR"
colnames(setStepChange)[3] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
```

```
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
```

```
        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
            ## correct direction
            svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
                i, 1]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
                1]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
            ## correct direction
            svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
            ## correct direction
            svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
    svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
    svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
    svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
    svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
        TRUE)
    svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
        )

    ##### Prepare output data #####
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
        predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
        stepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "PredictedClose_1"
    colnames(outputData)[5] <- "PredictedChange_1"
    colnames(outputData)[6] <- "PredictedStepChange_1"
    colnames(outputData)[7] <- "PredictedChange_Close_1"
    colnames(outputData)[8] <- "PredictedStepChange_Close_1"
    colnames(outputData)[9] <- "PredictedClose_Error_1"
    colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
    colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
        error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
        close.error2.r2, svr.stepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Close_MSE"
    colnames(errorData)[5] <- "Change_MSE"
    colnames(errorData)[6] <- "StepChange_MSE"
    colnames(errorData)[7] <- "Close_R2"
    colnames(errorData)[8] <- "Change_R2"
    colnames(errorData)[9] <- "StepChange_R2"

    errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
        stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
        change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
        direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
    errorByDirectionData <- as.data.frame(errorByDirectionData)
    colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
    colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
    colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
    colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
    colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
    colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
    colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
    colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
    colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
```

```
308
errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
313 # find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
318 }
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
323 # Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
328 lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
333 # smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
##### ====== End ====== #####
```

PredictPrice13 : EMD from closing price

```
##### ====== Begin ====== #####
# clear screen
3 rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
8 # arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
13 outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
18 # read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
23 dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

28 # init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
33     dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
38 dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
43     previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

48 # init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
53     if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
58 }
dataDirection <- as.matrix(dataDirection)

# EMD
```

```
library(EMD)
imf <- emd(dataClose[1:(numRecord-numTest)], 1:(numRecord-numTest))$imf
for (i in 1:numTest) {
    temp <- emd(dataClose[1:(numRecord-numTest+i)], 1:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
```

```
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
```

```
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
```

```
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice14 : EMD from closing change

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments ######
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data ######
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# EMD
library(EMD)
imf <- emd(dataChange[2:(numRecord-numTest)], 2:(numRecord-numTest))$imf
imf <- rbind(rep(NA, length(imf[1,])), imf)
for (i in 1:numTest) {
```

```
    temp <- emd(dataChange[2:(numRecord-numTest+i)], 2:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
```

```
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
```

```
                1]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
            ## correct direction
            svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
            ## correct direction
            svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
    svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
    svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
    svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
    svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
        TRUE)
    svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
        )

    ##### Prepare output data #####
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
        predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
        stepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "PredictedClose_1"
    colnames(outputData)[5] <- "PredictedChange_1"
    colnames(outputData)[6] <- "PredictedStepChange_1"
    colnames(outputData)[7] <- "PredictedChange_Close_1"
    colnames(outputData)[8] <- "PredictedStepChange_Close_1"
    colnames(outputData)[9] <- "PredictedClose_Error_1"
    colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
    colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
        error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
        close.error2.r2, svr.stepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Close_MSE"
    colnames(errorData)[5] <- "Change_MSE"
    colnames(errorData)[6] <- "StepChange_MSE"
    colnames(errorData)[7] <- "Close_R2"
    colnames(errorData)[8] <- "Change_R2"
    colnames(errorData)[9] <- "StepChange_R2"

    errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
        stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
        change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
        direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
    errorByDirectionData <- as.data.frame(errorByDirectionData)
    colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
    colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
    colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
    colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
    colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
    colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
    colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
    colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
    colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

    errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
    write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

    ##### Plot Graph #####
    # find min, max value to define the y range of graph
    minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
```

```
maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice15 : EMD from tick change

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# EMD
library(EMD)
imf <- emd(dataStepChange[2:(numRecord-numTest)], 2:(numRecord-numTest))$imf
imf <- rbind(rep(NA, length(imf[1,])), imf)
for (i in 1:numTest) {
    temp <- emd(dataStepChange[2:(numRecord-numTest+i)], 2:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
```

```
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i, 1]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i, 1])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1]) / dataClose[
        length(dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i, 1])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, ra.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
```

```
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i, 1] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i, 1] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i, 1] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i, 1]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i,
            1]) / dataClose[length(dataClose) - numTest + i]
    }
```

```
        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
            ## correct direction
            svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
                close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }

        if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
            ## correct direction
            svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
    }
    svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
    svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
    svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
    svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
    svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
        TRUE)
    svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
        )

    ##### Prepare output data #####
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
        predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
        stepChange_close.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "PredictedClose_1"
    colnames(outputData)[5] <- "PredictedChange_1"
    colnames(outputData)[6] <- "PredictedStepChange_1"
    colnames(outputData)[7] <- "PredictedChange_Close_1"
    colnames(outputData)[8] <- "PredictedStepChange_Close_1"
    colnames(outputData)[9] <- "PredictedClose_Error_1"
    colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
    colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

    errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
        error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
        close.error2.r2, svr.stepChange_close.error2.r2)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Close_MSE"
    colnames(errorData)[5] <- "Change_MSE"
    colnames(errorData)[6] <- "StepChange_MSE"
    colnames(errorData)[7] <- "Close_R2"
    colnames(errorData)[8] <- "Change_R2"
    colnames(errorData)[9] <- "StepChange_R2"

    errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
        stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
        change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
        direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
    errorByDirectionData <- as.data.frame(errorByDirectionData)
    colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
    colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
    colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
    colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
    colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
    colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
    colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
    colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
    colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"


    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

    errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
    write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

    ##### Plot Graph #####
    # find min, max value to define the y range of graph
    minValue <- min(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
    maxValue <- max(outputData[,1], svr.close.predict[,1], svr.change_close.predict, svr.stepChange_close.predict)
    if (is.finite(minValue) == FALSE) {
        minValue = 0
```

```
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ====== End ====== ######
```

PredictPrice16 : MACD

```
###### ====== Begin ====== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
macdColumnIndex = grep("MACD.26.12.9.", colnames(inputData))
signalColumnIndex = grep("Signal", colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataMACD <- inputData[,macdColumnIndex]
dataSignal <- inputData[,signalColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# init variable to store # of svm to use in each row
dataModel <- rep(NA, numRecord)
# calculate model
# #1 - macd plus, above signal
# #2 - macd minus, above signal
# #3 - macd plus, below signal
# #4 - macd minus, below signal
```

```
    for (i in 1:numRecord)
{
    if (dataMACD[i] > dataSignal[i]) {
        if (dataMACD[i] >= 0) {
            dataModel[i] = 1
        } else {
            dataModel[i] = 2
        }
    } else {
        if (dataMACD[i] >= 0) {
            dataModel[i] = 3
        } else {
            dataModel[i] = 4
        }
    }
}

##### Prepare Train Data #####
# pair of data: (current, next)
# current: remove last row
# next: remove first row
pairClose <- as.data.frame(cbind(dataClose[-numRecord], dataClose[-1]))
colnames(pairClose)[1] <- "Close"
colnames(pairClose)[2] <- "PredictedClose"
pairChange <- as.data.frame(cbind(dataChange[-numRecord], dataChange[-1]))
colnames(pairChange)[1] <- "Change"
colnames(pairChange)[2] <- "PredictedChange"
pairStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataStepChange[-1]))
colnames(pairStepChange)[1] <- "StepChange"
colnames(pairStepChange)[2] <- "PredictedStepChange"
# split into 2 variable: train, test
pairClose.train <- as.data.frame(pairClose[1:(length(pairClose[,1]) - numTest),])
pairClose.test <- as.data.frame(pairClose[-1:-(length(pairClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
pairChange.train <- as.data.frame(pairChange[2:(length(pairChange[,1]) - numTest),])
pairChange.test <- as.data.frame(pairChange[-1:-(length(pairChange[,1]) - numTest),])
pairStepChange.train <- as.data.frame(pairStepChange[2:(length(pairStepChange[,1]) - numTest),])
pairStepChange.test <- as.data.frame(pairStepChange[-1:-(length(pairStepChange[,1]) - numTest),])

# split into 2 variable: train, test
pairClose.train <- as.data.frame(pairClose[1:(length(pairClose[,1]) - numTest),])
pairClose.test <- as.data.frame(pairClose[-1:-(length(pairClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
pairChange.train <- as.data.frame(pairChange[2:(length(pairChange[,1]) - numTest),])
pairChange.test <- as.data.frame(pairChange[-1:-(length(pairChange[,1]) - numTest),])
pairStepChange.train <- as.data.frame(pairStepChange[2:(length(pairStepChange[,1]) - numTest),])
pairStepChange.test <- as.data.frame(pairStepChange[-1:-(length(pairStepChange[,1]) - numTest),])

# split train data into 4 groups
# #1 - macd plus, above signal
# #2 - macd minus, above signal
# #3 - macd plus, below signal
# #4 - macd minus, below signal
pairClose.train.model.1 <- data.frame(Close=NA, PredictedClose=NA)
pairClose.train.model.2 <- data.frame(Close=NA, PredictedClose=NA)
pairClose.train.model.3 <- data.frame(Close=NA, PredictedClose=NA)
pairClose.train.model.4 <- data.frame(Close=NA, PredictedClose=NA)
pairChange.train.model.1 <- data.frame(Change=NA, PredictedChange=NA)
pairChange.train.model.2 <- data.frame(Change=NA, PredictedChange=NA)
pairChange.train.model.3 <- data.frame(Change=NA, PredictedChange=NA)
pairChange.train.model.4 <- data.frame(Change=NA, PredictedChange=NA)
pairStepChange.train.model.1 <- data.frame(StepChange=NA, PredictedStepChange=NA)
pairStepChange.train.model.2 <- data.frame(StepChange=NA, PredictedStepChange=NA)
pairStepChange.train.model.3 <- data.frame(StepChange=NA, PredictedStepChange=NA)
pairStepChange.train.model.4 <- data.frame(StepChange=NA, PredictedStepChange=NA)
for (i in 1:length(pairClose.train[,1]))
{
    if (dataModel[i] == 1) {
        pairClose.train.model.1 <- rbind(pairClose.train.model.1, pairClose.train[i,])
        pairChange.train.model.1 <- rbind(pairChange.train.model.1, pairChange.train[i,])
        pairStepChange.train.model.1 <- rbind(pairStepChange.train.model.1, pairStepChange.train[i,])
    } else if (dataModel[i] == 2) {
        pairClose.train.model.2 <- rbind(pairClose.train.model.2, pairClose.train[i,])
        pairChange.train.model.2 <- rbind(pairChange.train.model.2, pairChange.train[i,])
        pairStepChange.train.model.2 <- rbind(pairStepChange.train.model.2, pairStepChange.train[i,])
    } else if (dataModel[i] == 3) {
        pairClose.train.model.3 <- rbind(pairClose.train.model.3, pairClose.train[i,])
        pairChange.train.model.3 <- rbind(pairChange.train.model.3, pairChange.train[i,])
        pairStepChange.train.model.3 <- rbind(pairStepChange.train.model.3, pairStepChange.train[i,])
    } else if (dataModel[i] == 4) {
        pairClose.train.model.4 <- rbind(pairClose.train.model.4, pairClose.train[i,])
        pairChange.train.model.4 <- rbind(pairChange.train.model.4, pairChange.train[i,])
        pairStepChange.train.model.4 <- rbind(pairStepChange.train.model.4, pairStepChange.train[i,])
    }
}

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model.1 <- ksvm(PredictedClose~., data = pairClose.train.model.1, type = svmType, kernel = kernelType)
svr.close.model.2 <- ksvm(PredictedClose~., data = pairClose.train.model.2, type = svmType, kernel = kernelType)
svr.close.model.3 <- ksvm(PredictedClose~., data = pairClose.train.model.3, type = svmType, kernel = kernelType)
svr.close.model.4 <- ksvm(PredictedClose~., data = pairClose.train.model.4, type = svmType, kernel = kernelType)
svr.change.model.1 <- ksvm(PredictedChange~., data = pairChange.train.model.1, type = svmType, kernel = kernelType)
```

```
svr.change.model.2 <- ksvm(PredictedChange~., data = pairChange.train.model.2, type = svmType, kernel = kernelType)
svr.change.model.3 <- ksvm(PredictedChange~., data = pairChange.train.model.3, type = svmType, kernel = kernelType)
svr.change.model.4 <- ksvm(PredictedChange~., data = pairChange.train.model.4, type = svmType, kernel = kernelType)
svr.stepChange.model.1 <- ksvm(PredictedStepChange~., data = pairStepChange.train.model.1, type = svmType, kernel =
    kernelType)
svr.stepChange.model.2 <- ksvm(PredictedStepChange~., data = pairStepChange.train.model.2, type = svmType, kernel =
    kernelType)
svr.stepChange.model.3 <- ksvm(PredictedStepChange~., data = pairStepChange.train.model.3, type = svmType, kernel =
    kernelType)
svr.stepChange.model.4 <- ksvm(PredictedStepChange~., data = pairStepChange.train.model.4, type = svmType, kernel =
    kernelType)

##### Predict 1 ahead #####
# select model based on the last model of train data
lastModel <- dataModel[length(pairClose.train)]
if (lastModel == 1) {
    svr.close.model <- svr.close.model.1
    svr.change.model <- svr.change.model.1
    svr.stepChange.model <- svr.stepChange.model.1
} else if (lastModel == 2) {
    svr.close.model <- svr.close.model.2
    svr.change.model <- svr.change.model.2
    svr.stepChange.model <- svr.stepChange.model.2
} else if (lastModel == 3) {
    svr.close.model <- svr.close.model.3
    svr.change.model <- svr.change.model.3
    svr.stepChange.model <- svr.stepChange.model.3
} else if (lastModel == 4) {
    svr.close.model <- svr.close.model.4
    svr.change.model <- svr.change.model.4
    svr.stepChange.model <- svr.stepChange.model.4
}
# close
svr.close.predict <- predict(svr.close.model, pairClose.test)
# change
svr.change.predict <- predict(svr.change.model, pairChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, pairStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
```

```
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
            ]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
```

```
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        } else {
            ## wrong direction
            svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
                stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
        }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
```

```
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

PredictPrice17 : EMA Change

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
ema10ColumnIndex = grep("EMAV.C.10.", colnames(inputData))
ema60ColumnIndex = grep("EMAV.C.60.", colnames(inputData))
ema250ColumnIndex = grep("EMAV.C.250.", colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataEMA10 <- inputData[,ema10ColumnIndex]
dataEMA60 <- inputData[,ema60ColumnIndex]
dataEMA250 <- inputData[,ema250ColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(1, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# init variable to store ema change
dataEMA10Change <- rep(NA, numRecord)
dataEMA60Change <- rep(NA, numRecord)
dataEMA250Change <- rep(NA, numRecord)
# calculate ema change
for (i in 2:numRecord)
{
    dataEMA10Change[i] <- dataEMA10[i] - dataEMA10[i-1]
    dataEMA60Change[i] <- dataEMA60[i] - dataEMA60[i-1]
    dataEMA250Change[i] <- dataEMA250[i] - dataEMA250[i-1]
}

##### Prepare Train Data #####
# set of data: (current, ema10, ema60, ema250, next)
# current, ema10, ema60, ema250: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord],
```

```
        dataEMA250Change[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "EMA10Change"
colnames(setClose)[3] <- "EMA60Change"
colnames(setClose)[4] <- "EMA250Change"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord],
        dataEMA250Change[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "EMA10Change"
colnames(setChange)[3] <- "EMA60Change"
colnames(setChange)[4] <- "EMA250Change"
colnames(setChange)[5] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-
        numRecord], dataEMA250Change[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "EMA10Change"
colnames(setStepChange)[3] <- "EMA60Change"
colnames(setStepChange)[4] <- "EMA250Change"
colnames(setStepChange)[5] <- "PredictedStepChange"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
# change, stepChange need to exclude the first row as its value is NA in first row
setChange.train <- as.data.frame(setChange[2:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[2:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )

##### Predict 1 ahead #####
# close
svr.close.predict <- predict(svr.close.model, setClose.test)
# change
svr.change.predict <- predict(svr.change.model, setChange.test)
# stepChange
svr.stepChange.predict <- predict(svr.stepChange.model, setStepChange.test)

##### Calculate PredictedChange_Close, PredictedStepChange_Close #####
# change
svr.change_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.change_close.predict[i] <- dataClose[length(dataClose) - numTest + i - 1] + svr.change.predict[i]
}
# stepChange
svr.stepChange_close.predict <- rep(NA, numTest)
for (i in 1:numTest) {
    svr.stepChange_close.predict[i] <- priceFromStep(stepFromPrice(dataClose[length(dataClose) - numTest + i - 1])
        + svr.stepChange.predict[i])
}

##### Calculate Error #####
close.average <- mean(dataClose[(length(dataClose) - numTest + 1):length(dataClose)])
close.sum2 <- rep(NA, numTest)
for (i in 1:numTest) {
    close.sum2[i] <- (dataClose[length(dataClose) - numTest + i] - close.average)^2
}
close.tot <- sum(close.sum2, na.rm = TRUE)
# close
svr.close.error <- rep(NA, numTest)
svr.close.error.mape <- vector()
svr.close.error2 <- rep(NA, numTest)
svr.close.error2.mse <- vector()
svr.close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i]) / dataClose[length
        (dataClose) - numTest + i]
    svr.close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.close.predict[i])^2
}
svr.close.error.mape[1] <- mean(svr.close.error, na.rm = TRUE)
svr.close.error2.mse[1] <- mean(svr.close.error2, na.rm = TRUE)
svr.close.error2.r2[1] <- 1 - sum(svr.close.error2, na.rm = TRUE) / close.tot
# change
svr.change_close.error <- rep(NA, numTest)
svr.change_close.error.mape <- vector()
svr.change_close.error2 <- rep(NA, numTest)
svr.change_close.error2.mse <- vector()
svr.change_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.change_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i]) /
        dataClose[length(dataClose) - numTest + i]
    svr.change_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.change_close.predict[i])^2
}
svr.change_close.error.mape[1] <- mean(svr.change_close.error, na.rm = TRUE)
svr.change_close.error2.mse[1] <- mean(svr.change_close.error2, na.rm = TRUE)
svr.change_close.error2.r2[1] <- 1 - sum(svr.change_close.error2, na.rm = TRUE) / close.tot
```

```
# stepChange
svr.stepChange_close.error <- rep(NA, numTest)
svr.stepChange_close.error.mape <- vector()
svr.stepChange_close.error2 <- rep(NA, numTest)
svr.stepChange_close.error2.mse <- vector()
svr.stepChange_close.error2.r2 <- vector()
for (i in 1:numTest) {
    svr.stepChange_close.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[
        i]) / dataClose[length(dataClose) - numTest + i]
    svr.stepChange_close.error2[i] <- (dataClose[length(dataClose) - numTest + i] - svr.stepChange_close.predict[i
        ])^2
}
svr.stepChange_close.error.mape[1] <- mean(svr.stepChange_close.error, na.rm = TRUE)
svr.stepChange_close.error2.mse[1] <- mean(svr.stepChange_close.error2, na.rm = TRUE)
svr.stepChange_close.error2.r2[1] <- 1 - sum(svr.stepChange_close.error2, na.rm = TRUE) / close.tot

##### Calculate Direction #####
dataDirection.close <- rep(NA, numTest)
dataDirection.change <- rep(NA, numTest)
dataDirection.stepChange <- rep(NA, numTest)
for (i in 1:numTest) {
    if (svr.close.predict[i] - dataClose[length(dataClose) - numTest + i] < 0) {
        dataDirection.close[i] <- -1
    } else {
        dataDirection.close[i] <- 1
    }
    if (svr.change.predict[i] < 0) {
        dataDirection.change[i] <- -1
    } else {
        dataDirection.change[i] <- 1
    }
    if (svr.stepChange.predict[i] < 0) {
        dataDirection.stepChange[i] <- -1
    } else {
        dataDirection.stepChange[i] <- 1
    }
}

##### Calculate Direction Error #####
dataDirection.close.error <- rep(NA, numTest)
dataDirection.close.error.mape <- vector()
dataDirection.change.error <- rep(NA, numTest)
dataDirection.change.error.mape <- vector()
dataDirection.stepChange.error <- rep(NA, numTest)
dataDirection.stepChange.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection.close[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.close.error[i] <- 0
    } else {
        dataDirection.close.error[i] <- 1
    }
    if (dataDirection.change[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.change.error[i] <- 0
    } else {
        dataDirection.change.error[i] <- 1
    }
    if (dataDirection.stepChange[i] == dataDirection[length(dataDirection) - numTest + i]) {
        dataDirection.stepChange.error[i] <- 0
    } else {
        dataDirection.stepChange.error[i] <- 1
    }
}
dataDirection.close.error.mape[1] <- mean(dataDirection.close.error, na.rm = TRUE)
dataDirection.change.error.mape[1] <- mean(dataDirection.change.error, na.rm = TRUE)
dataDirection.stepChange.error.mape[1] <- mean(dataDirection.stepChange.error, na.rm = TRUE)

##### Calculate Error by correctness of Direction #####
svr.close.direction_correct.error <- rep(NA, numTest)
svr.close.direction_correct.error.mape <- vector()
svr.close.direction_wrong.error <- rep(NA, numTest)
svr.close.direction_wrong.error.mape <- vector()
svr.change_close.direction_correct.error <- rep(NA, numTest)
svr.change_close.direction_correct.error.mape <- vector()
svr.change_close.direction_wrong.error <- rep(NA, numTest)
svr.change_close.direction_wrong.error.mape <- vector()
svr.stepChange_close.direction_correct.error <- rep(NA, numTest)
svr.stepChange_close.direction_correct.error.mape <- vector()
svr.stepChange_close.direction_wrong.error <- rep(NA, numTest)
svr.stepChange_close.direction_wrong.error.mape <- vector()
for (i in 1:numTest) {
    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.close[i]) {
        ## correct direction
        svr.close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[
            i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.close.predict[i
            ]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.change[i]) {
        ## correct direction
        svr.change_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
```

```
    } else {
        ## wrong direction
        svr.change_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.change_
            close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }

    if (dataDirection[length(dataDirection) - numTest + i] == dataDirection.stepChange[i]) {
        ## correct direction
        svr.stepChange_close.direction_correct.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    } else {
        ## wrong direction
        svr.stepChange_close.direction_wrong.error[i] <- abs(dataClose[length(dataClose) - numTest + i] - svr.
            stepChange_close.predict[i]) / dataClose[length(dataClose) - numTest + i]
    }
}
svr.close.direction_correct.error.mape[1] <- mean(svr.close.direction_correct.error, na.rm = TRUE)
svr.close.direction_wrong.error.mape[1] <- mean(svr.close.direction_wrong.error, na.rm = TRUE)
svr.change_close.direction_correct.error.mape[1] <- mean(svr.change_close.direction_correct.error, na.rm = TRUE)
svr.change_close.direction_wrong.error.mape[1] <- mean(svr.change_close.direction_wrong.error, na.rm = TRUE)
svr.stepChange_close.direction_correct.error.mape[1] <- mean(svr.stepChange_close.direction_correct.error, na.rm =
    TRUE)
svr.stepChange_close.direction_wrong.error.mape[1] <- mean(svr.stepChange_close.direction_wrong.error, na.rm = TRUE
    )

##### Prepare output data #####
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], svr.close.predict, svr.change.predict, svr.stepChange.
    predict, svr.change_close.predict, svr.stepChange_close.predict, svr.close.error, svr.change_close.error, svr.
    stepChange_close.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "PredictedClose_1"
colnames(outputData)[5] <- "PredictedChange_1"
colnames(outputData)[6] <- "PredictedStepChange_1"
colnames(outputData)[7] <- "PredictedChange_Close_1"
colnames(outputData)[8] <- "PredictedStepChange_Close_1"
colnames(outputData)[9] <- "PredictedClose_Error_1"
colnames(outputData)[10] <- "PredictedChange_Close_Error_1"
colnames(outputData)[11] <- "PredictedStepChange_Close_Error_1"

errorData <- cbind(svr.close.error.mape, svr.change_close.error.mape, svr.stepChange_close.error.mape, svr.close.
    error2.mse, svr.change_close.error2.mse, svr.stepChange_close.error2.mse, svr.close.error2.r2, svr.change_
    close.error2.r2, svr.stepChange_close.error2.r2)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Close_MSE"
colnames(errorData)[5] <- "Change_MSE"
colnames(errorData)[6] <- "StepChange_MSE"
colnames(errorData)[7] <- "Close_R2"
colnames(errorData)[8] <- "Change_R2"
colnames(errorData)[9] <- "StepChange_R2"

errorByDirectionData <- cbind(dataDirection.close.error.mape, dataDirection.change.error.mape, dataDirection.
    stepChange.error.mape, svr.close.direction_correct.error.mape, svr.close.direction_wrong.error.mape, svr.
    change_close.direction_correct.error.mape, svr.change_close.direction_wrong.error.mape, svr.stepChange_close.
    direction_correct.error.mape, svr.stepChange_close.direction_wrong.error.mape)
errorByDirectionData <- as.data.frame(errorByDirectionData)
colnames(errorByDirectionData)[1] <- "Close_Direction_MAPE"
colnames(errorByDirectionData)[2] <- "Change_Direction_MAPE"
colnames(errorByDirectionData)[3] <- "StepChange_Direction_MAPE"
colnames(errorByDirectionData)[4] <- "Close_Correct_MAPE"
colnames(errorByDirectionData)[5] <- "Close_Wrong_MAPE"
colnames(errorByDirectionData)[6] <- "Change_Correct_MAPE"
colnames(errorByDirectionData)[7] <- "Change_Wrong_MAPE"
colnames(errorByDirectionData)[8] <- "StepChange_Correct_MAPE"
colnames(errorByDirectionData)[9] <- "StepChange_Wrong_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)

errorByDirectionFileName <- paste(outputDir, "ErrorByDirection.txt", sep = "/")
write.csv(errorByDirectionData, file = errorByDirectionFileName, row.names = FALSE, col.names = TRUE)

##### Plot Graph #####
# find min, max value to define the y range of graph
minValue <- min(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
maxValue <- max(outputData[,1], svr.close.predict, svr.change_close.predict, svr.stepChange_close.predict)
if (is.finite(minValue) == FALSE) {
    minValue = 0
}
if (is.finite(maxValue) == FALSE) {
    maxValue = 400
}
```

```
outputPNGName <- paste(outputDir, "/", "Predicted_", i, ".png", sep = "")
# Start PNG device driver to save output to PNG
png(filename = outputPNGName, height = 1200, width = 1600, bg = "white")
# Graph using blue points overlayed by a line
plot(outputData[,1], type = "o", col = "black", lty = 1, lwd = 2, ylim = c(minValue, maxValue))
lines(svr.close.predict, type = "l", col = "black", lty = 2, lwd = 2)
lines(svr.change_close.predict, type = "l", col = "black", lty = 3, lwd = 2)
lines(svr.stepChange_close.predict, type = "l", col = "black", lty = 4, lwd = 2)
# Create box around plot
box()
# Create a legend in the top-left corner that is slightly
# smaller and has no border
legend("topleft", legend = c("Close", "PredictedClose", "PredictedChangeClose", "PredictedStepChangeClose"), cex =
    0.8, col = "black", lty = 1:4, lwd = 2, bty = "n");
# Turn off device driver (to flush output to png)
dev.off()
###### ===== End ===== ######
```

Direction Prediction Script

PredictDirection1 : Time Series

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments ######
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data ######
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data ######
# set of data: (index, current)
setClose <- as.data.frame(cbind(1:numRecord, dataClose))
setChange <- as.data.frame(cbind(1:numRecord, dataChange))
```

```
setStepChange <- as.data.frame(cbind(1:numRecord, dataStepChange))
setDirection <- as.data.frame(cbind(1:numRecord, dataDirection))

colnames(setClose)[1] <- "Index"
colnames(setChange)[1] <- "Index"
colnames(setStepChange)[1] <- "Index"
colnames(setDirection)[1] <- "Index"

colnames(setClose)[2] <- "PredictedClose"
colnames(setChange)[2] <- "PredictedChange"
colnames(setStepChange)[2] <- "PredictedStepChange"
colnames(setDirection)[2] <- "PredictedDirection"
# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict numStep ahead #####
# close
# create dummy column
svr.close.predict <- rep(NA, numTest)
tempSet <- setClose.test
for (i in 1:numStep)
{
    temp <- predict(svr.close.model, tempSet)
    svr.close.predict <- cbind(svr.close.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.close.predict <- as.data.frame(svr.close.predict[,-1])

# change
# create dummy column
svr.change.predict <- rep(NA, numTest)
tempSet <- setChange.test
for (i in 1:numStep)
{
    temp <- predict(svr.change.model, tempSet)
    svr.change.predict <- cbind(svr.change.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.change.predict <- as.data.frame(svr.change.predict[,-1])

# stepChange
# create dummy column
svr.stepChange.predict <- rep(NA, numTest)
tempSet <- setStepChange.test
for (i in 1:numStep)
{
    temp <- predict(svr.stepChange.model, tempSet)
    svr.stepChange.predict <- cbind(svr.stepChange.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.stepChange.predict <- as.data.frame(svr.stepChange.predict[,-1])

# direction
# create dummy column
svr.direction.predict <- rep(NA, numTest)
tempSet <- setDirection.test
for (i in 1:numStep)
{
    temp <- predict(svr.direction.model, tempSet)
    svr.direction.predict <- cbind(svr.direction.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.direction.predict <- as.data.frame(svr.direction.predict[,-1])

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
```

```
    # change
    for (i in 1:numStep)
    {
        svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
167     colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    }
    # stepChange
    for (i in 1:numStep)
    {
172     svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
            , i])
        colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    }
    # direction
    for (i in 1:numStep)
177 {
        svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
            ])
        colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    }

182 ##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
    # close
    # create dummy column
    svr.close_direction.predict <- rep(NA, numTest)
    prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
187 for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
192         if (svr.close.predict[j, i] - prevClose[j] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
            }
197     }
        svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
        # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
            )
        prevClose <- c(NA, svr.close.predict[,i])
    }
202 # remove dummy column
    svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

    # change
    # create dummy column
207 svr.change_direction.predict <- rep(NA, numTest)
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
212     {
            if (svr.change.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
217         }
        }
        svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
    }
    # remove dummy column
222 svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.predict <- rep(NA, numTest)
227 for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
232         if (svr.stepChange.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
            }
237     }
        svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
    }
    # remove dummy column
    svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])
242
    ##### Readjust PredictedDirection #####
    for (i in 1:numStep)
    {
        for (j in i:numTest)
247     {
            if (svr.direction.predict[j, i] < 0) {
                svr.direction.predict[j, i] <- -1
            } else {
                svr.direction.predict[j, i] <- 1
252         }
        }
    }
```

```
##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
```

```
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

###### Write data to csv file ######
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
###### ===== End ===== ######
```

PredictDirection2 : Historical data

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")

###### Read command line arguments ######
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])

###### Prepare Input Data ######
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
```

```
        if (dataChange[i] < 0) {
            dataDirection[i] <- -1
        } else {
            dataDirection[i] <- 1
        }
    }
    dataDirection <- as.matrix(dataDirection)

    ##### Prepare Train Data #####
    # set of data: (prev_1, prev_2, ..., prev_n, next)
    # prev_i: remove first i-1 rows and last n-i+1 rows
    # next: remove first n rows
    # create dummy column
    setClose <- rep(NA, numRecord - numUse)
    setChange <- rep(NA, numRecord - numUse)
    setStepChange <- rep(NA, numRecord - numUse)
    setDirection <- rep(NA, numRecord - numUse)
    for (i in 1:numUse)
    {
        setClose <- cbind(setClose, dataClose[i:(numRecord-numUse+i-1)])
        setChange <- cbind(setChange, dataChange[i:(numRecord-numUse+i-1)])
        setStepChange <- cbind(setStepChange, dataStepChange[i:(numRecord-numUse+i-1)])
        setDirection <- cbind(setDirection, dataDirection[i:(numRecord-numUse+i-1)])
    }
    # remove dummy column, and append "next" column
    setClose <- as.data.frame(cbind(setClose[,-1], dataClose[(numUse+1):numRecord]))
    setChange <- as.data.frame(cbind(setChange[,-1], dataChange[(numUse+1):numRecord]))
    setStepChange <- as.data.frame(cbind(setStepChange[,-1], dataStepChange[(numUse+1):numRecord]))
    setDirection <- as.data.frame(cbind(setDirection[,-1], dataDirection[(numUse+1):numRecord]))

    for (i in 1:numUse)
    {
        colnames(setClose)[i] <- paste("Close", i, sep = "_")
        colnames(setChange)[i] <- paste("Change", i, sep = "_")
        colnames(setStepChange)[i] <- paste("StepChange", i, sep = "_")
        colnames(setDirection)[i] <- paste("Direction", i, sep = "_")
    }
    colnames(setClose)[numUse+1] <- "PredictedClose"
    colnames(setChange)[numUse+1] <- "PredictedChange"
    colnames(setStepChange)[numUse+1] <- "PredictedStepChange"
    colnames(setDirection)[numUse+1] <- "PredictedDirection"
    # split into 2 variable: train, test
    setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
    setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
    setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
    setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
    setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
    setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
    setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
    setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

    ##### Train SVM #####
    # load library
    library(kernlab)
    # create svm model with model xxx~., i.e., xxx column is product of the others columns
    svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
    svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
    svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
        )
    svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

    ##### Predict numStep ahead #####
    # close
    # create dummy column
    svr.close.predict <- rep(NA, numTest)
    tempSet <- setClose.test
    for (i in 1:numStep)
    {
        temp <- predict(svr.close.model, tempSet)
        svr.close.predict <- cbind(svr.close.predict, temp)
        tempSet <- as.data.frame(cbind(tempSet[,2:numUse], temp))
        for (j in 1:numUse)
        {
            colnames(tempSet)[j] <- paste("Close", j, sep = "_")
        }
    }
    # remove dummy column
    svr.close.predict <- as.data.frame(svr.close.predict[,-1])

    # change
    # create dummy column
    svr.change.predict <- rep(NA, numTest)
    tempSet <- setChange.test
    for (i in 1:numStep)
    {
        temp <- predict(svr.change.model, tempSet)
        svr.change.predict <- cbind(svr.change.predict, temp)
        tempSet <- as.data.frame(cbind(tempSet[,2:numUse], temp))
        for (j in 1:numUse)
        {
            colnames(tempSet)[j] <- paste("Change", j, sep = "_")
        }
    }
    # remove dummy column
    svr.change.predict <- as.data.frame(svr.change.predict[,-1])
```

```
# stepChange
# create dummy column
svr.stepChange.predict <- rep(NA, numTest)
tempSet <- setStepChange.test
for (i in 1:numStep)
{
    temp <- predict(svr.stepChange.model, tempSet)
    svr.stepChange.predict <- cbind(svr.stepChange.predict, temp)
    tempSet <- as.data.frame(cbind(tempSet[,2:numUse], temp))
    for (j in 1:numUse)
    {
        colnames(tempSet)[j] <- paste("StepChange", j, sep = "_")
    }
}
# remove dummy column
svr.stepChange.predict <- as.data.frame(svr.stepChange.predict[,-1])

# direction
# create dummy column
svr.direction.predict <- rep(NA, numTest)
tempSet <- setDirection.test
for (i in 1:numStep)
{
    temp <- predict(svr.direction.model, tempSet)
    svr.direction.predict <- cbind(svr.direction.predict, temp)
    tempSet <- as.data.frame(cbind(tempSet[,2:numUse], temp))
    for (j in 1:numUse)
    {
        colnames(tempSet)[j] <- paste("Direction", j, sep = "_")
    }
}
# remove dummy column
svr.direction.predict <- as.data.frame(svr.direction.predict[,-1])

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
```

```
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
```

```
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ====== End ====== #####
```

PredictDirection3 : EMA

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
```

```
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
ema10ColumnIndex = grep("EMAV.C.10.", colnames(inputData))
ema60ColumnIndex = grep("EMAV.C.60.", colnames(inputData))
ema250ColumnIndex = grep("EMAV.C.250.", colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataEMA10 <- inputData[,ema10ColumnIndex]
dataEMA60 <- inputData[,ema60ColumnIndex]
dataEMA250 <- inputData[,ema250ColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, ema10, ema60, ema250, next)
# current, ema10, ema60, ema250: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord], dataEMA250[-
    numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "EMA10"
colnames(setClose)[3] <- "EMA60"
colnames(setClose)[4] <- "EMA250"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord], dataEMA250[-
    numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "EMA10"
colnames(setChange)[3] <- "EMA60"
colnames(setChange)[4] <- "EMA250"
colnames(setChange)[5] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord],
    dataEMA250[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "EMA10"
colnames(setStepChange)[3] <- "EMA60"
colnames(setStepChange)[4] <- "EMA250"
colnames(setStepChange)[5] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataEMA10[-numRecord], dataEMA60[-numRecord],
    dataEMA250[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "EMA10"
colnames(setDirection)[3] <- "EMA60"
colnames(setDirection)[4] <- "EMA250"
colnames(setDirection)[5] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
```

```
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
```

```
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
```

```
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ===== End ===== #####
```

PredictDirection4 : Volume

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
```

```
volColumnIndex = grep(paste("^", "Vol", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataVol <- inputData[,volColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, vol, next)
# current, vol: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataVol[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "Vol"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataVol[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "Vol"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataVol[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "Vol"
colnames(setStepChange)[3] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataVol[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "Vol"
colnames(setDirection)[3] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
```

```
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
```

```
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
```

```
306 for (i in 1:numStep) {
      colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
      colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
      colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
      colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
311   colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
      colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
      colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
      colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
      colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
316   colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
      colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
    }
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
        svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
        direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
        change_direction.error, svr.stepChange_direction.error, svr.direction.error)
    outputData <- as.data.frame(outputData)
321 colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "Direction"

326 errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
        mape, svr.direction.error.mape)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
331 colnames(errorData)[4] <- "Direction_MAPE"

    ###### Write data to csv file ######
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
336 # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
341 write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
    ###### ===== End ===== ######
```

PredictDirection5 : Buy-Sell Volume

```
    ###### ===== Begin ===== ######
    # clear screen
  3 rm(list=ls())
    # include function
    source("Util.R")

    ###### Read command line arguments ######
  8 # arguments: inputFileName outputDir svmType kernelType numUse
    # read arguments
    args <- commandArgs(TRUE)
    # parse args
    inputFileName = as.character(args[1])
 13 outputDir = as.character(args[2])
    svmType = as.character(args[3])
    kernelType = as.character(args[4])
    numUse = as.integer(args[5])
    numUse <- 1
 18
    ###### Prepare Input Data ######
    # read a file in csv format
    inputData <- read.csv(inputFileName)
    # find the column index of targetColumnName
 23 targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
    buyVolColumnIndex = grep(paste("Buy.Vol", "\$", sep = ""), colnames(inputData))
    sellVolColumnIndex = grep(paste("Sell.Vol", "\$", sep = ""), colnames(inputData))
    # assign data from targetColumnIndex to new variable
    dataClose <- inputData[,targetColumnIndex]
 28 dataBuyVol <- inputData[,buyVolColumnIndex]
    dataSellVol <- inputData[,sellVolColumnIndex]
    numRecord <- length(dataClose)
    numTest <- floor(numRecord * 0.3)
    numTrain <- numRecord - numTest
 33 numStep <- 1

    # init variable to store price change
    dataChange <- rep(NA, numRecord)
    # calculate price change
 38 for (i in 2:numRecord)
    {
        dataChange[i] <- dataClose[i] - dataClose[i-1]
    }
    dataChange <- as.matrix(dataChange)
 43
    # init variable to store step change
    dataStepChange <- rep(NA, numRecord)
    # calculate step change
    for (i in 2:numRecord)
 48 {
        step <- stepFromPrice(dataClose[i])
```

```
        previousStep <- stepFromPrice(dataClose[i-1])
        dataStepChange[i] <- step - previousStep
    }
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, buyVol, sellVol, next)
# current, buyVol, sellVol: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord], dataClose
    [-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "BuyVol"
colnames(setClose)[3] <- "SellVol"
colnames(setClose)[4] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "BuyVol"
colnames(setChange)[3] <- "SellVol"
colnames(setChange)[4] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "BuyVol"
colnames(setStepChange)[3] <- "SellVol"
colnames(setStepChange)[4] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord],
    dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "BuyVol"
colnames(setDirection)[3] <- "SellVol"
colnames(setDirection)[4] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))


##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
```

```
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
```

```
            if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.close_direction.error <- cbind(svr.close_direction.error, temp)
        svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

    # change
    # create dummy column
    svr.change_direction.error <- rep(NA, numTest)
    svr.change_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.change_direction.error <- cbind(svr.change_direction.error, temp)
        svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.error <- rep(NA, numTest)
    svr.stepChange_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
        svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

    # direction
    # create dummy column
    svr.direction.error <- rep(NA, numTest)
    svr.direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.direction.error <- cbind(svr.direction.error, temp)
        svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
    # remove dummy column
    svr.direction.error <- as.data.frame(svr.direction.error[,-1])

    ##### Prepare output data #####
    for (i in 1:numStep) {
        colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
        colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
        colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
        colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
        colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
        colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
        colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
        colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
        colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
        colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
        colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
    }
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
          dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
```

```
        svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
        direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
        change_direction.error, svr.stepChange_direction.error, svr.direction.error)
    outputData <- as.data.frame(outputData)
328 colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "Direction"

333 errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
        mape, svr.direction.error.mape)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
338 colnames(errorData)[4] <- "Direction_MAPE"

    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
343 # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
348 write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
    ##### ====== End ====== #####
```

PredictDirection6 : CMF (Chaikin Money Flow)

```
1   ##### ====== Begin ====== #####
    # clear screen
    rm(list=ls())
    # include function
    source("Util.R")
6
    ##### Read command line arguments #####
    # arguments: inputFileName outputDir svmType kernelType numUse
    # read arguments
    args <- commandArgs(TRUE)
11  # parse args
    inputFileName = as.character(args[1])
    outputDir = as.character(args[2])
    svmType = as.character(args[3])
    kernelType = as.character(args[4])
16  numUse = as.integer(args[5])
    numUse <- 1

    ##### Prepare Input Data #####
    # read a file in csv format
21  inputData <- read.csv(inputFileName)
    # find the column index of targetColumnName
    targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
    cmfColumnIndex = grep(paste("^", "CMF", "\$", sep = ""), colnames(inputData))
    # assign data from targetColumnIndex to new variable
26  dataClose <- inputData[,targetColumnIndex]
    dataCMF <- inputData[,cmfColumnIndex]
    numRecord <- length(dataClose)
    numTest <- floor(numRecord * 0.3)
    numTrain <- numRecord - numTest
31  numStep <- 1

    # init variable to store price change
    dataChange <- rep(NA, numRecord)
    # calculate price change
36  for (i in 2:numRecord)
    {
        dataChange[i] <- dataClose[i] - dataClose[i-1]
    }
    dataChange <- as.matrix(dataChange)
41
    # init variable to store step change
    dataStepChange <- rep(NA, numRecord)
    # calculate step change
    for (i in 2:numRecord)
46  {
        step <- stepFromPrice(dataClose[i])
        previousStep <- stepFromPrice(dataClose[i-1])
        dataStepChange[i] <- step - previousStep
    }
51  dataStepChange <- as.matrix(dataStepChange)

    # init variable to store direction
    dataDirection <- rep(NA, numRecord)
    # calculate direction
56  for (i in 2:numRecord)
    {
        if (dataChange[i] < 0) {
            dataDirection[i] <- -1
        } else {
61          dataDirection[i] <- 1
        }
    }
    dataDirection <- as.matrix(dataDirection)
```

```
##### Prepare Train Data #####
# set of data: (current, cmf, next)
# current, cmf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataCMF[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "CMF"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataCMF[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "CMF"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataCMF[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "CMF"
colnames(setStepChange)[3] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataCMF[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "CMF"
colnames(setDirection)[3] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
```

```
        }
        svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
        # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
            )
        prevClose <- c(NA, svr.close.predict[,i])
    }
    # remove dummy column
    svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

    # change
    # create dummy column
    svr.change_direction.predict <- rep(NA, numTest)
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.change.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
            }
        }
        svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
    }
    # remove dummy column
    svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.predict <- rep(NA, numTest)
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.stepChange.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
            }
        }
        svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
    }
    # remove dummy column
    svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

    ##### Readjust PredictedDirection #####
    for (i in 1:numStep)
    {
        for (j in i:numTest)
        {
            if (svr.direction.predict[j, i] < 0) {
                svr.direction.predict[j, i] <- -1
            } else {
                svr.direction.predict[j, i] <- 1
            }
        }
    }

    ##### Calculate Error #####
    # close
    # create dummy column
    svr.close_direction.error <- rep(NA, numTest)
    svr.close_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.close_direction.error <- cbind(svr.close_direction.error, temp)
        svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

    # change
    # create dummy column
    svr.change_direction.error <- rep(NA, numTest)
    svr.change_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
```

```
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

###### Prepare output data ######
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
      dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
      svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
      direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
      change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
      mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

###### Write data to csv file ######
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
```

```
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ====== End ====== #####
```

PredictDirection7 : MOF (Money Flow)

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
mofColumnIndex = grep(paste("^", "MOF", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataMOF <- inputData[,mofColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, mof, next)
# current, mof: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataMOF[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "MOF"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataMOF[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "MOF"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataMOF[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "MOF"
colnames(setStepChange)[3] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataMOF[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "MOF"
colnames(setDirection)[3] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
```

```
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType)
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1), i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating)
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
```

```
    # remove dummy column
    svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.predict <- rep(NA, numTest)
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.stepChange.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
            }
        }
        svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
    }
    # remove dummy column
    svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

    ##### Readjust PredictedDirection #####
    for (i in 1:numStep)
    {
        for (j in i:numTest)
        {
            if (svr.direction.predict[j, i] < 0) {
                svr.direction.predict[j, i] <- -1
            } else {
                svr.direction.predict[j, i] <- 1
            }
        }
    }

    ##### Calculate Error #####
    # close
    # create dummy column
    svr.close_direction.error <- rep(NA, numTest)
    svr.close_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.close_direction.error <- cbind(svr.close_direction.error, temp)
        svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

    # change
    # create dummy column
    svr.change_direction.error <- rep(NA, numTest)
    svr.change_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.change_direction.error <- cbind(svr.change_direction.error, temp)
        svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.error <- rep(NA, numTest)
    svr.stepChange_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
```

```
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ===== End ===== #####
```

PredictDirection8 : OBV (On Balance Volume)

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
```

```
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
obvColumnIndex = grep(paste("^", "OBV", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataOBV <- inputData[,obvColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, obv, next)
# current, obv: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataOBV[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "OBV"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataOBV[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "OBV"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataOBV[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "OBV"
colnames(setStepChange)[3] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataOBV[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "OBV"
colnames(setDirection)[3] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))
```

```
##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
```

```
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
```

```
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ===== End ===== #####
```

PredictDirection9 : RSI (Relative Strength Index)

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
rsiColumnIndex = grep(paste("^", "RSI.14.", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataRSI <- inputData[,rsiColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
```

```
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, rsi, next)
# current, rsi: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataRSI[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "RSI"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataRSI[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "RSI"
colnames(setChange)[3] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataRSI[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "RSI"
colnames(setStepChange)[3] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataRSI[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "RSI"
colnames(setDirection)[3] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
```

```
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
```

```
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
```

```
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ====== End ====== #####
```

PredictDirection10 : ADX (Average Directional Index), DIP (Directional Index Plus), and DIM (Directional Index Minus)

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
adxColumnIndex = grep(paste("^", "ADX", "\$", sep = ""), colnames(inputData))
dipColumnIndex = grep(paste("^", "DI.", "\$", sep = ""), colnames(inputData))
dimColumnIndex = grep(paste("^", "DI..1", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataADX <- inputData[,adxColumnIndex]
dataDIP <- inputData[,dipColumnIndex]
dataDIM <- inputData[,dimColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)
```

```
##### Prepare Train Data #####
# set of data: (current, adx, dip, dim, next)
# current, adx, dip, dim: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "ATR"
colnames(setClose)[3] <- "DIP"
colnames(setClose)[4] <- "DIM"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "ATR"
colnames(setChange)[3] <- "DIP"
colnames(setChange)[4] <- "DIM"
colnames(setChange)[5] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "ATR"
colnames(setStepChange)[3] <- "DIP"
colnames(setStepChange)[4] <- "DIM"
colnames(setStepChange)[5] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataADX[-numRecord], dataDIP[-numRecord], dataDIM[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "ATR"
colnames(setDirection)[3] <- "DIP"
colnames(setDirection)[4] <- "DIM"
colnames(setDirection)[5] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType)
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1), i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
```

```
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])
```

```
# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

###### Prepare output data ######
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
```

```
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ===== End ===== #####
```

PredictDirection11 : ATR (Average True Range)

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
atrColumnIndex = grep(paste("^", "ATR", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataATR <- inputData[,atrColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

##### Prepare Train Data #####
# set of data: (current, atr, next)
# current, atr: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataATR[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "ATR"
colnames(setClose)[3] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataATR[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "ATR"
colnames(setChange)[3] <- "PredictedChange"
```

```
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataATR[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "ATR"
colnames(setStepChange)[3] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataATR[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "ATR"
colnames(setDirection)[3] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
```

```
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
```

```
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

###### Prepare output data ######
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

###### Write data to csv file ######
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
###### ===== End ===== ######
```

PredictDirection12 : EMD of closing price

```
###### ===== Begin ===== ######
# clear screen
rm(list=ls())
# include function
source("Util.R")
```

```
##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# EMD
library(EMD)
imf <- emd(dataClose[1:(numRecord-numTest)], 1:(numRecord-numTest))$imf
for (i in 1:numTest) {
    temp <- emd(dataClose[1:(numRecord-numTest+i)], 1:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

setDirection <- as.data.frame(cbind(dataDirection[-numRecord], imf[-numRecord,], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
for (i in 1:length(imf[1,])) {
    colnames(setDirection)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setDirection)[length(setDirection[1,])] <- "PredictedDirection"
```

```
103
    # split into 2 variable: train, test
    setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
    setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
    setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
108 setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
    setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
    setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
    setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
    setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])
113
    ##### Train SVM#####
    # load library
    library(kernlab)
    # create svm model with model xxx~., i.e., xxx column is product of the others columns
118 svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
    svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
    svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
        )
    svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

123 ##### Predict 1 ahead #####
    # close
    svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
    # change
    svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
128 # stepChange
    svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
    # direction
    svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

133 ##### Reformat predicted data #####
    # close
    for (i in 1:numStep)
    {
        svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
138     colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    }
    # change
    for (i in 1:numStep)
    {
143     svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
        colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    }
    # stepChange
    for (i in 1:numStep)
148 {
        svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
            , i])
        colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    }
    # direction
153 for (i in 1:numStep)
    {
        svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
            ])
        colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    }
158
    ##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
    # close
    # create dummy column
    svr.close_direction.predict <- rep(NA, numTest)
163 prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
168     {
            if (svr.close.predict[j, i] - prevClose[j] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
173         }
        }
        svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
        # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
            )
        prevClose <- c(NA, svr.close.predict[,i])
178 }
    # remove dummy column
    svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

    # change
183 # create dummy column
    svr.change_direction.predict <- rep(NA, numTest)
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
188     for (j in i:numTest)
        {
            if (svr.change.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
193             temp[j] <- 1
```

```
            }
        }
        svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
    }
    # remove dummy column
    svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.predict <- rep(NA, numTest)
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.stepChange.predict[j, i] < 0) {
                temp[j] <- -1
            } else {
                temp[j] <- 1
            }
        }
        svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
    }
    # remove dummy column
    svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

    ###### Readjust PredictedDirection ######
    for (i in 1:numStep)
    {
        for (j in i:numTest)
        {
            if (svr.direction.predict[j, i] < 0) {
                svr.direction.predict[j, i] <- -1
            } else {
                svr.direction.predict[j, i] <- 1
            }
        }
    }

    ###### Calculate Error ######
    # close
    # create dummy column
    svr.close_direction.error <- rep(NA, numTest)
    svr.close_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.close_direction.error <- cbind(svr.close_direction.error, temp)
        svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

    # change
    # create dummy column
    svr.change_direction.error <- rep(NA, numTest)
    svr.change_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.change_direction.error <- cbind(svr.change_direction.error, temp)
        svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

    # stepChange
    # create dummy column
    svr.stepChange_direction.error <- rep(NA, numTest)
    svr.stepChange_direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
```

```
            } else {
                temp[j] <- 1
            }
        }
        svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
        svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
    # remove dummy column
    svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

    # direction
    # create dummy column
    svr.direction.error <- rep(NA, numTest)
    svr.direction.error.mape <- vector()
    for (i in 1:numStep)
    {
        temp <- rep(NA, numTest)
        for (j in i:numTest)
        {
            if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
                temp[j] <- 0
            } else {
                temp[j] <- 1
            }
        }
        svr.direction.error <- cbind(svr.direction.error, temp)
        svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
    # remove dummy column
    svr.direction.error <- as.data.frame(svr.direction.error[,-1])

    ##### Prepare output data #####
    for (i in 1:numStep) {
        colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
        colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
        colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
        colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
        colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
        colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
        colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
        colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
        colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
        colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
        colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
    }
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
          dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
          svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
          direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
          change_direction.error, svr.stepChange_direction.error, svr.direction.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "Direction"

    errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
          mape, svr.direction.error.mape)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Direction_MAPE"

    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
    ##### ====== End ====== #####
```

PredictDirection13 : EMD of closing change

```
    ##### ====== Begin ====== #####
    # clear screen
    rm(list=ls())
    # include function
    source("Util.R")

    ##### Read command line arguments #####
    # arguments: inputFileName outputDir svmType kernelType numUse
    # read arguments
    args <- commandArgs(TRUE)
    # parse args
    inputFileName = as.character(args[1])
    outputDir = as.character(args[2])
    svmType = as.character(args[3])
    kernelType = as.character(args[4])
```

```
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# EMD
library(EMD)
imf <- emd(dataChange[2:(numRecord-numTest)], 2:(numRecord-numTest))$imf
imf <- rbind(rep(NA, length(imf[1,])), imf)
for (i in 1:numTest) {
    temp <- emd(dataChange[2:(numRecord-numTest+i)], 2:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

setDirection <- as.data.frame(cbind(dataDirection[-numRecord], imf[-numRecord,], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
for (i in 1:length(imf[1,])) {
    colnames(setDirection)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setDirection)[length(setDirection[1,])] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
```

```
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType)
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
```

```
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
```

```
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ====== End ====== #####
```

PredictDirection14 : EMD of tick change

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
```

```
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# EMD
library(EMD)
imf <- emd(dataStepChange[2:(numRecord-numTest)], 2:(numRecord-numTest))$imf
imf <- rbind(rep(NA, length(imf[1,])), imf)
for (i in 1:numTest) {
    temp <- emd(dataStepChange[2:(numRecord-numTest+i)], 2:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

setDirection <- as.data.frame(cbind(dataDirection[-numRecord], imf[-numRecord,], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
for (i in 1:length(imf[1,])) {
    colnames(setDirection)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setDirection)[length(setDirection[1,])] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
```

```
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
```

```
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
```

```
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ===== End ===== #####
```

PredictDirection15 : EMD of direction

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
```

```
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# EMD
library(EMD)
imf <- emd(dataDirection[2:(numRecord-numTest)], 2:(numRecord-numTest))$imf
imf <- rbind(rep(NA, length(imf[1,])), imf)
for (i in 1:numTest) {
    temp <- emd(dataDirection[2:(numRecord-numTest+i)], 2:(numRecord-numTest+i), max.imf=length(imf[1,]))$imf
    imf <- rbind(imf, temp[length(temp[,1]),])
}

##### Prepare Train Data #####
# set of data: (current, imf, next)
# current, imf: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], imf[-numRecord,], dataClose[-1]))
colnames(setClose)[1] <- "Close"
for (i in 1:length(imf[1,])) {
    colnames(setClose)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setClose)[length(setClose[1,])] <- "PredictedClose"

setChange <- as.data.frame(cbind(dataChange[-numRecord], imf[-numRecord,], dataChange[-1]))
colnames(setChange)[1] <- "Change"
for (i in 1:length(imf[1,])) {
    colnames(setChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setChange)[length(setChange[1,])] <- "PredictedChange"

setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], imf[-numRecord,], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
for (i in 1:length(imf[1,])) {
    colnames(setStepChange)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setStepChange)[length(setStepChange[1,])] <- "PredictedStepChange"

setDirection <- as.data.frame(cbind(dataDirection[-numRecord], imf[-numRecord,], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
for (i in 1:length(imf[1,])) {
    colnames(setDirection)[1+i] <- paste("IMF", i, sep = "_")
}
colnames(setDirection)[length(setDirection[1,])] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
```

```
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
```

```
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
```

```
            }
        }
        svr.direction.error <- cbind(svr.direction.error, temp)
        svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
    }
    svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
    # remove dummy column
    svr.direction.error <- as.data.frame(svr.direction.error[,-1])

    ##### Prepare output data #####
    for (i in 1:numStep) {
        colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
        colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
        colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
        colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
        colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
        colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
        colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
        colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
        colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
        colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
        colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
    }
    outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
        dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
        svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
        direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
        change_direction.error, svr.stepChange_direction.error, svr.direction.error)
    outputData <- as.data.frame(outputData)
    colnames(outputData)[1] <- "Close"
    colnames(outputData)[2] <- "Change"
    colnames(outputData)[3] <- "StepChange"
    colnames(outputData)[4] <- "Direction"

    errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
        mape, svr.direction.error.mape)
    errorData <- as.data.frame(errorData)
    colnames(errorData)[1] <- "Close_MAPE"
    colnames(errorData)[2] <- "Change_MAPE"
    colnames(errorData)[3] <- "StepChange_MAPE"
    colnames(errorData)[4] <- "Direction_MAPE"

    ##### Write data to csv file #####
    # create output directory if not exist
    dir.create(outputDir, recursive = TRUE)
    # write output data to file (in csv format)
    outputFileName <- paste(outputDir, "Result.txt", sep = "/")
    write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

    errorFileName <- paste(outputDir, "Error.txt", sep = "/")
    write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
    ##### ===== End ===== #####
```

PredictDirection16 : MACD

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "$", sep = ""), colnames(inputData))
macdColumnIndex = grep("MACD.26.12.9.", colnames(inputData))
signalColumnIndex = grep("Signal", colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataMACD <- inputData[,macdColumnIndex]
dataSignal <- inputData[,signalColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
```

```
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# init variable to store # of svm to use in each row
dataModel <- rep(NA, numRecord)
# calculate model
# #1 - macd plus, above signal
# #2 - macd minus, above signal
# #3 - macd plus, below signal
# #4 - macd minus, below signal
for (i in 1:numRecord)
{
    if (dataMACD[i] >= dataSignal[i]) {
        if (dataMACD[i] >= 0) {
            dataModel[i] = 1
        } else {
            dataModel[i] = 2
        }
    } else {
        if (dataMACD[i] >= 0) {
            dataModel[i] = 3
        } else {
            dataModel[i] = 4
        }
    }
}

##### Prepare Train Data #####
# set of data: (current, next)
# current: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

# split train data into 4 groups
# #1 - macd plus, above signal
# #2 - macd minus, above signal
# #3 - macd plus, below signal
# #4 - macd minus, below signal
setClose.train.model.1 <- data.frame(Close = NA, PredictedClose = NA)
setClose.train.model.2 <- data.frame(Close = NA, PredictedClose = NA)
setClose.train.model.3 <- data.frame(Close = NA, PredictedClose = NA)
setClose.train.model.4 <- data.frame(Close = NA, PredictedClose = NA)
setChange.train.model.1 <- data.frame(Change = NA, PredictedChange = NA)
setChange.train.model.2 <- data.frame(Change = NA, PredictedChange = NA)
setChange.train.model.3 <- data.frame(Change = NA, PredictedChange = NA)
setChange.train.model.4 <- data.frame(Change = NA, PredictedChange = NA)
setStepChange.train.model.1 <- data.frame(StepChange = NA, PredictedStepChange = NA)
setStepChange.train.model.2 <- data.frame(StepChange = NA, PredictedStepChange = NA)
setStepChange.train.model.3 <- data.frame(StepChange = NA, PredictedStepChange = NA)
setStepChange.train.model.4 <- data.frame(StepChange = NA, PredictedStepChange = NA)
setDirection.train.model.1 <- data.frame(Direction = NA, PredictedDirection = NA)
```

```
setDirection.train.model.2 <- data.frame(Direction = NA, PredictedDirection = NA)
setDirection.train.model.3 <- data.frame(Direction = NA, PredictedDirection = NA)
setDirection.train.model.4 <- data.frame(Direction = NA, PredictedDirection = NA)
for (i in 1:length(setClose.train[,1]))
{
    if (dataModel[i] == 1) {
        setClose.train.model.1 <- rbind(setClose.train.model.1, setClose.train[i,])
        setChange.train.model.1 <- rbind(setChange.train.model.1, setChange.train[i,])
        setStepChange.train.model.1 <- rbind(setStepChange.train.model.1, setStepChange.train[i,])
        setDirection.train.model.1 <- rbind(setDirection.train.model.1, setDirection.train[i,])
    } else if (dataModel[i] == 2) {
        setClose.train.model.2 <- rbind(setClose.train.model.2, setClose.train[i,])
        setChange.train.model.2 <- rbind(setChange.train.model.2, setChange.train[i,])
        setStepChange.train.model.2 <- rbind(setStepChange.train.model.2, setStepChange.train[i,])
        setDirection.train.model.2 <- rbind(setDirection.train.model.2, setDirection.train[i,])
    } else if (dataModel[i] == 3) {
        setClose.train.model.3 <- rbind(setClose.train.model.3, setClose.train[i,])
        setChange.train.model.3 <- rbind(setChange.train.model.3, setChange.train[i,])
        setStepChange.train.model.3 <- rbind(setStepChange.train.model.3, setStepChange.train[i,])
        setDirection.train.model.3 <- rbind(setDirection.train.model.3, setDirection.train[i,])
    } else if (dataModel[i] == 4) {
        setClose.train.model.4 <- rbind(setClose.train.model.4, setClose.train[i,])
        setChange.train.model.4 <- rbind(setChange.train.model.4, setChange.train[i,])
        setStepChange.train.model.4 <- rbind(setStepChange.train.model.4, setStepChange.train[i,])
        setDirection.train.model.4 <- rbind(setDirection.train.model.4, setDirection.train[i,])
    }
}

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model.1 <- ksvm(PredictedClose~., data = setClose.train.model.1, type = svmType, kernel = kernelType)
svr.close.model.2 <- ksvm(PredictedClose~., data = setClose.train.model.2, type = svmType, kernel = kernelType)
svr.close.model.3 <- ksvm(PredictedClose~., data = setClose.train.model.3, type = svmType, kernel = kernelType)
svr.close.model.4 <- ksvm(PredictedClose~., data = setClose.train.model.4, type = svmType, kernel = kernelType)
svr.change.model.1 <- ksvm(PredictedChange~., data = setChange.train.model.1, type = svmType, kernel = kernelType)
svr.change.model.2 <- ksvm(PredictedChange~., data = setChange.train.model.2, type = svmType, kernel = kernelType)
svr.change.model.3 <- ksvm(PredictedChange~., data = setChange.train.model.3, type = svmType, kernel = kernelType)
svr.change.model.4 <- ksvm(PredictedChange~., data = setChange.train.model.4, type = svmType, kernel = kernelType)
svr.stepChange.model.1 <- ksvm(PredictedStepChange~., data = setStepChange.train.model.1, type = svmType, kernel =
    kernelType)
svr.stepChange.model.2 <- ksvm(PredictedStepChange~., data = setStepChange.train.model.2, type = svmType, kernel =
    kernelType)
svr.stepChange.model.3 <- ksvm(PredictedStepChange~., data = setStepChange.train.model.3, type = svmType, kernel =
    kernelType)
svr.stepChange.model.4 <- ksvm(PredictedStepChange~., data = setStepChange.train.model.4, type = svmType, kernel =
    kernelType)
svr.direction.model.1 <- ksvm(PredictedDirection~., data = setDirection.train.model.1, type = svmType, kernel =
    kernelType)
svr.direction.model.2 <- ksvm(PredictedDirection~., data = setDirection.train.model.2, type = svmType, kernel =
    kernelType)
svr.direction.model.3 <- ksvm(PredictedDirection~., data = setDirection.train.model.3, type = svmType, kernel =
    kernelType)
svr.direction.model.4 <- ksvm(PredictedDirection~., data = setDirection.train.model.4, type = svmType, kernel =
    kernelType)

##### Predict numStep ahead #####
# select model based on the last model of train data
lastModel <- dataModel[length(setClose.train)]
if (lastModel == 1) {
    svr.close.model <- svr.close.model.1
    svr.change.model <- svr.change.model.1
    svr.stepChange.model <- svr.stepChange.model.1
    svr.direction.model <- svr.direction.model.1
} else if (lastModel == 2) {
    svr.close.model <- svr.close.model.2
    svr.change.model <- svr.change.model.2
    svr.stepChange.model <- svr.stepChange.model.2
    svr.direction.model <- svr.direction.model.2
} else if (lastModel == 3) {
    svr.close.model <- svr.close.model.3
    svr.change.model <- svr.change.model.3
    svr.stepChange.model <- svr.stepChange.model.3
    svr.direction.model <- svr.direction.model.3
} else if (lastModel == 4) {
    svr.close.model <- svr.close.model.4
    svr.change.model <- svr.change.model.4
    svr.stepChange.model <- svr.stepChange.model.4
    svr.direction.model <- svr.direction.model.4
}
# close
# create dummy column
svr.close.predict <- rep(NA, numTest)
tempSet <- setClose.test
for (i in 1:numStep)
{
    temp <- predict(svr.close.model, tempSet)
    svr.close.predict <- cbind(svr.close.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.close.predict <- as.data.frame(svr.close.predict[,-1])
```

```
# change
# create dummy column
svr.change.predict <- rep(NA, numTest)
tempSet <- setChange.test
for (i in 1:numStep)
{
    temp <- predict(svr.change.model, tempSet)
    svr.change.predict <- cbind(svr.change.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.change.predict <- as.data.frame(svr.change.predict[,-1])

# stepChange
# create dummy column
svr.stepChange.predict <- rep(NA, numTest)
tempSet <- setStepChange.test
for (i in 1:numStep)
{
    temp <- predict(svr.stepChange.model, tempSet)
    svr.stepChange.predict <- cbind(svr.stepChange.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.stepChange.predict <- as.data.frame(svr.stepChange.predict[,-1])

# direction
# create dummy column
svr.direction.predict <- rep(NA, numTest)
tempSet <- setDirection.test
for (i in 1:numStep)
{
    temp <- predict(svr.direction.model, tempSet)
    svr.direction.predict <- cbind(svr.direction.predict, temp)
    tempSet <- as.data.frame(tempSet[,1] + rep(1, length(tempSet[,1])))
    colnames(tempSet)[1] <- "Index"
}
# remove dummy column
svr.direction.predict <- as.data.frame(svr.direction.predict[,-1])

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
```

```
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

###### Readjust PredictedDirection ######
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

###### Calculate Error ######
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])
```

```
# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ====== End ====== #####
```

PredictDirection17 : EMA Change

```
##### ====== Begin ====== #####
# clear screen
rm(list=ls())
```

```
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
ema10ColumnIndex = grep("EMAV.C.10.", colnames(inputData))
ema60ColumnIndex = grep("EMAV.C.60.", colnames(inputData))
ema250ColumnIndex = grep("EMAV.C.250.", colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataEMA10 <- inputData[,ema10ColumnIndex]
dataEMA60 <- inputData[,ema60ColumnIndex]
dataEMA250 <- inputData[,ema250ColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# init variable to store ema change
dataEMA10Change <- rep(NA, numRecord)
dataEMA60Change <- rep(NA, numRecord)
dataEMA250Change <- rep(NA, numRecord)
# calculate ema change
for (i in 2:numRecord)
{
    dataEMA10Change[i] <- dataEMA10[i] - dataEMA10[i-1]
    dataEMA60Change[i] <- dataEMA60[i] - dataEMA60[i-1]
    dataEMA250Change[i] <- dataEMA250[i] - dataEMA250[i-1]
}

##### Prepare Train Data #####
# set of data: (current, ema10, ema60, ema250, next)
# current, ema10, ema60, ema250: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord],
    dataEMA250Change[-numRecord], dataClose[-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "EMA10Change"
colnames(setClose)[3] <- "EMA60Change"
colnames(setClose)[4] <- "EMA250Change"
colnames(setClose)[5] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord],
    dataEMA250Change[-numRecord], dataChange[-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "EMA10Change"
colnames(setChange)[3] <- "EMA60Change"
colnames(setChange)[4] <- "EMA250Change"
colnames(setChange)[5] <- "PredictedChange"
```

```
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord], dataEMA250Change[-numRecord], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "EMA10Change"
colnames(setStepChange)[3] <- "EMA60Change"
colnames(setStepChange)[4] <- "EMA250Change"
colnames(setStepChange)[5] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord], dataEMA250Change[-numRecord], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "EMA10Change"
colnames(setDirection)[3] <- "EMA60Change"
colnames(setDirection)[4] <- "EMA250Change"
colnames(setDirection)[5] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM#####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType)
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1), i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i])
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating)
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
```

```
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
```

```
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

###### Prepare output data ######
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

###### Write data to csv file ######
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
###### ====== End ====== ######
```

PredictDirection18 : EMA Change, Volume, and Buy-Sell Volume

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
numUse = as.integer(args[5])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
ema10ColumnIndex = grep("EMAV.C.10.", colnames(inputData))
ema60ColumnIndex = grep("EMAV.C.60.", colnames(inputData))
ema250ColumnIndex = grep("EMAV.C.250.", colnames(inputData))
volColumnIndex = grep(paste("^", "Vol", "\$", sep = ""), colnames(inputData))
buyVolColumnIndex = grep(paste("Buy.Vol", "\$", sep = ""), colnames(inputData))
sellVolColumnIndex = grep(paste("Sell.Vol", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataEMA10 <- inputData[,ema10ColumnIndex]
dataEMA60 <- inputData[,ema60ColumnIndex]
dataEMA250 <- inputData[,ema250ColumnIndex]
dataVol <- inputData[,volColumnIndex]
dataBuyVol <- inputData[,buyVolColumnIndex]
dataSellVol <- inputData[,sellVolColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)

# init variable to store ema change
dataEMA10Change <- rep(NA, numRecord)
dataEMA60Change <- rep(NA, numRecord)
dataEMA250Change <- rep(NA, numRecord)
# calculate ema change
for (i in 2:numRecord)
{
    dataEMA10Change[i] <- dataEMA10[i] - dataEMA10[i-1]
    dataEMA60Change[i] <- dataEMA60[i] - dataEMA60[i-1]
    dataEMA250Change[i] <- dataEMA250[i] - dataEMA250[i-1]
}

##### Prepare Train Data #####
# set of data: (current, ema10, ema60, ema250, vol, buyVol, sellVol, next)
# current, ema10, ema60, ema250, vol, buyVol, sellVol: remove last row
# next: remove first row
setClose <- as.data.frame(cbind(dataClose[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord],
    dataEMA250Change[-numRecord], dataVol[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord], dataClose
    [-1]))
colnames(setClose)[1] <- "Close"
colnames(setClose)[2] <- "EMA10Change"
```

```
colnames(setClose)[3] <- "EMA60Change"
colnames(setClose)[4] <- "EMA250Change"
colnames(setClose)[5] <- "Vol"
colnames(setClose)[6] <- "BuyVol"
colnames(setClose)[7] <- "SellVol"
colnames(setClose)[8] <- "PredictedClose"
setChange <- as.data.frame(cbind(dataChange[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-numRecord],
    dataEMA250Change[-numRecord], dataVol[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord], dataChange
    [-1]))
colnames(setChange)[1] <- "Change"
colnames(setChange)[2] <- "EMA10Change"
colnames(setChange)[3] <- "EMA60Change"
colnames(setChange)[4] <- "EMA250Change"
colnames(setChange)[5] <- "Vol"
colnames(setChange)[6] <- "BuyVol"
colnames(setChange)[7] <- "SellVol"
colnames(setChange)[8] <- "PredictedChange"
setStepChange <- as.data.frame(cbind(dataStepChange[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-
    numRecord], dataEMA250Change[-numRecord], dataVol[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord
    ], dataStepChange[-1]))
colnames(setStepChange)[1] <- "StepChange"
colnames(setStepChange)[2] <- "EMA10Change"
colnames(setStepChange)[3] <- "EMA60Change"
colnames(setStepChange)[4] <- "EMA250Change"
colnames(setStepChange)[5] <- "Vol"
colnames(setStepChange)[6] <- "BuyVol"
colnames(setStepChange)[7] <- "SellVol"
colnames(setStepChange)[8] <- "PredictedStepChange"
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-
    numRecord], dataEMA250Change[-numRecord], dataVol[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord
    ], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "EMA10Change"
colnames(setDirection)[3] <- "EMA60Change"
colnames(setDirection)[4] <- "EMA250Change"
colnames(setDirection)[5] <- "Vol"
colnames(setDirection)[6] <- "BuyVol"
colnames(setDirection)[7] <- "SellVol"
colnames(setDirection)[8] <- "PredictedDirection"

# split into 2 variable: train, test
setClose.train <- as.data.frame(setClose[1:(length(setClose[,1]) - numTest),])
setClose.test <- as.data.frame(setClose[-1:-(length(setClose[,1]) - numTest),])
setChange.train <- as.data.frame(setChange[1:(length(setChange[,1]) - numTest),])
setChange.test <- as.data.frame(setChange[-1:-(length(setChange[,1]) - numTest),])
setStepChange.train <- as.data.frame(setStepChange[1:(length(setStepChange[,1]) - numTest),])
setStepChange.test <- as.data.frame(setStepChange[-1:-(length(setStepChange[,1]) - numTest),])
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
svr.close.model <- ksvm(PredictedClose~., data = setClose.train, type = svmType, kernel = kernelType)
svr.change.model <- ksvm(PredictedChange~., data = setChange.train, type = svmType, kernel = kernelType)
svr.stepChange.model <- ksvm(PredictedStepChange~., data = setStepChange.train, type = svmType, kernel = kernelType
    )
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType, kernel = kernelType)

##### Predict 1 ahead #####
# close
svr.close.predict <- as.data.frame(predict(svr.close.model, setClose.test))
# change
svr.change.predict <- as.data.frame(predict(svr.change.model, setChange.test))
# stepChange
svr.stepChange.predict <- as.data.frame(predict(svr.stepChange.model, setStepChange.test))
# direction
svr.direction.predict <- as.data.frame(predict(svr.direction.model, setDirection.test))

##### Reformat predicted data #####
# close
for (i in 1:numStep)
{
    svr.close.predict[,i] <- c(rep(NA, i-1), svr.close.predict[1:(length(svr.close.predict[,i])-i+1), i])
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
}
# change
for (i in 1:numStep)
{
    svr.change.predict[,i] <- c(rep(NA, i-1), svr.change.predict[1:(length(svr.change.predict[,i])-i+1), i])
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
}
# stepChange
for (i in 1:numStep)
{
    svr.stepChange.predict[,i] <- c(rep(NA, i-1), svr.stepChange.predict[1:(length(svr.stepChange.predict[,i])-i+1)
        , i])
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
}
# direction
for (i in 1:numStep)
{
    svr.direction.predict[,i] <- c(rep(NA, i-1), svr.direction.predict[1:(length(svr.direction.predict[,i])-i+1), i
        ])
```

```
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
}

##### Calculate PredictedClose_Direction, PredictedChange_Direction, PredictedStepChange_Direction #####
# close
# create dummy column
svr.close_direction.predict <- rep(NA, numTest)
prevClose <- dataClose[(length(dataClose)-numTest-1):(length(dataClose)-1)]
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close.predict[j, i] - prevClose[j] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.predict <- cbind(svr.close_direction.predict, temp)
    # shift down 1 row, to make it work for the first column (no need to subtract 1 from row index when calculating
        )
    prevClose <- c(NA, svr.close.predict[,i])
}
# remove dummy column
svr.close_direction.predict <- as.data.frame(svr.close_direction.predict[,-1])

# change
# create dummy column
svr.change_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.predict <- cbind(svr.change_direction.predict, temp)
}
# remove dummy column
svr.change_direction.predict <- as.data.frame(svr.change_direction.predict[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.predict <- rep(NA, numTest)
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange.predict[j, i] < 0) {
            temp[j] <- -1
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.predict <- cbind(svr.stepChange_direction.predict, temp)
}
# remove dummy column
svr.stepChange_direction.predict <- as.data.frame(svr.stepChange_direction.predict[,-1])

##### Readjust PredictedDirection #####
for (i in 1:numStep)
{
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] < 0) {
            svr.direction.predict[j, i] <- -1
        } else {
            svr.direction.predict[j, i] <- 1
        }
    }
}

##### Calculate Error #####
# close
# create dummy column
svr.close_direction.error <- rep(NA, numTest)
svr.close_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.close_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.close_direction.error <- cbind(svr.close_direction.error, temp)
```

```
    svr.close_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.close_direction.error.mape[numStep+1] <- mean(svr.close_direction.error, na.rm = TRUE)
# remove dummy column
svr.close_direction.error <- as.data.frame(svr.close_direction.error[,-1])

# change
# create dummy column
svr.change_direction.error <- rep(NA, numTest)
svr.change_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.change_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.change_direction.error <- cbind(svr.change_direction.error, temp)
    svr.change_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.change_direction.error.mape[numStep+1] <- mean(svr.change_direction.error, na.rm = TRUE)
# remove dummy column
svr.change_direction.error <- as.data.frame(svr.change_direction.error[,-1])

# stepChange
# create dummy column
svr.stepChange_direction.error <- rep(NA, numTest)
svr.stepChange_direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.stepChange_direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.stepChange_direction.error <- cbind(svr.stepChange_direction.error, temp)
    svr.stepChange_direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.stepChange_direction.error.mape[numStep+1] <- mean(svr.stepChange_direction.error, na.rm = TRUE)
# remove dummy column
svr.stepChange_direction.error <- as.data.frame(svr.stepChange_direction.error[,-1])

# direction
# create dummy column
svr.direction.error <- rep(NA, numTest)
svr.direction.error.mape <- vector()
for (i in 1:numStep)
{
    temp <- rep(NA, numTest)
    for (j in i:numTest)
    {
        if (svr.direction.predict[j, i] == dataDirection[length(dataDirection) - numTest + j]) {
            temp[j] <- 0
        } else {
            temp[j] <- 1
        }
    }
    svr.direction.error <- cbind(svr.direction.error, temp)
    svr.direction.error.mape[i] <- mean(temp, na.rm = TRUE)
}
svr.direction.error.mape[numStep+1] <- mean(svr.direction.error, na.rm = TRUE)
# remove dummy column
svr.direction.error <- as.data.frame(svr.direction.error[,-1])

##### Prepare output data #####
for (i in 1:numStep) {
    colnames(svr.close.predict)[i] <- paste("PredictedClose", i, sep = "_")
    colnames(svr.change.predict)[i] <- paste("PredictedChange", i, sep = "_")
    colnames(svr.stepChange.predict)[i] <- paste("PredictedStepChange", i, sep = "_")
    colnames(svr.close_direction.predict)[i] <- paste("PredictedClose_Direction", i, sep = "_")
    colnames(svr.change_direction.predict)[i] <- paste("PredictedChange_Direction", i, sep = "_")
    colnames(svr.stepChange_direction.predict)[i] <- paste("PredictedStepChange_Direction", i, sep = "_")
    colnames(svr.direction.predict)[i] <- paste("PredictedDirection", i, sep = "_")
    colnames(svr.close_direction.error)[i] <- paste("PredictedClose_Direction_Error", i, sep = "_")
    colnames(svr.change_direction.error)[i] <- paste("PredictedChange_Direction_Error", i, sep = "_")
    colnames(svr.stepChange_direction.error)[i] <- paste("PredictedStepChange_Direction_Error", i, sep = "_")
    colnames(svr.direction.error)[i] <- paste("PredictedDirection_Error", i, sep = "_")
}
outputData <- cbind(dataClose[-1:-(length(dataClose) - numTest)], dataChange[-1:-(length(dataChange) - numTest)],
    dataStepChange[-1:-(length(dataStepChange) - numTest)], dataDirection[-1:-(length(dataDirection) - numTest)],
    svr.close.predict, svr.change.predict, svr.stepChange.predict, svr.close_direction.predict, svr.change_
    direction.predict, svr.stepChange_direction.predict, svr.direction.predict, svr.close_direction.error, svr.
    change_direction.error, svr.stepChange_direction.error, svr.direction.error)
outputData <- as.data.frame(outputData)
colnames(outputData)[1] <- "Close"
colnames(outputData)[2] <- "Change"
colnames(outputData)[3] <- "StepChange"
```

```
colnames(outputData)[4] <- "Direction"

errorData <- cbind(svr.close_direction.error.mape, svr.change_direction.error.mape, svr.stepChange_direction.error.
    mape, svr.direction.error.mape)
errorData <- as.data.frame(errorData)
colnames(errorData)[1] <- "Close_MAPE"
colnames(errorData)[2] <- "Change_MAPE"
colnames(errorData)[3] <- "StepChange_MAPE"
colnames(errorData)[4] <- "Direction_MAPE"

##### Write data to csv file #####
# create output directory if not exist
dir.create(outputDir, recursive = TRUE)
# write output data to file (in csv format)
outputFileName <- paste(outputDir, "Result.txt", sep = "/")
write.csv(outputData, file = outputFileName, row.names = FALSE, col.names = TRUE)

errorFileName <- paste(outputDir, "Error.txt", sep = "/")
write.csv(errorData, file = errorFileName, row.names = FALSE, col.names = TRUE)
##### ===== End ===== #####
```

Combined Prediction Script

PredictCombine1 : EMA Change, Volume, and Buy-Sell Volume for Direction, and Volume for Price

```
##### ===== Begin ===== #####
# clear screen
rm(list=ls())
# include function
source("Util.R")

##### Read command line arguments #####
# arguments: inputFileName outputDir svmType kernelType numUse
# read arguments
args <- commandArgs(TRUE)
# parse args
inputFileName = as.character(args[1])
outputDir = as.character(args[2])
svmType = as.character(args[3])
kernelType = as.character(args[4])
svmType_direction = as.character(args[5])
kernelType_direction = as.character(args[6])
numUse = as.integer(args[7])
numUse <- 1

##### Prepare Input Data #####
# read a file in csv format
inputData <- read.csv(inputFileName)
# find the column index of targetColumnName
targetColumnIndex = grep(paste(".Close", "\$", sep = ""), colnames(inputData))
ema10ColumnIndex = grep("EMAV.C.10.", colnames(inputData))
ema60ColumnIndex = grep("EMAV.C.60.", colnames(inputData))
ema250ColumnIndex = grep("EMAV.C.250.", colnames(inputData))
volColumnIndex = grep(paste("^", "Vol", "\$", sep = ""), colnames(inputData))
buyVolColumnIndex = grep(paste("Buy.Vol", "\$", sep = ""), colnames(inputData))
sellVolColumnIndex = grep(paste("Sell.Vol", "\$", sep = ""), colnames(inputData))
# assign data from targetColumnIndex to new variable
dataClose <- inputData[,targetColumnIndex]
dataEMA10 <- inputData[,ema10ColumnIndex]
dataEMA60 <- inputData[,ema60ColumnIndex]
dataEMA250 <- inputData[,ema250ColumnIndex]
dataVol <- inputData[,volColumnIndex]
dataBuyVol <- inputData[,buyVolColumnIndex]
dataSellVol <- inputData[,sellVolColumnIndex]
numRecord <- length(dataClose)
numTest <- floor(numRecord * 0.3)
numTrain <- numRecord - numTest
numStep <- 1

dataVol.train <- dataVol[1:(length(dataVol) - numTest)]
dataVol.test <- dataVol[-1:-(length(dataVol) - numTest)]

# init variable to store price change
dataChange <- rep(NA, numRecord)
# calculate price change
for (i in 2:numRecord)
{
    dataChange[i] <- dataClose[i] - dataClose[i-1]
}
dataChange <- as.matrix(dataChange)
dataChange.train <- dataChange[1:(length(dataChange) - numTest)]
dataChange.test <- dataChange[-1:-(length(dataChange) - numTest)]

# init variable to store step change
dataStepChange <- rep(NA, numRecord)
```

```
# calculate step change
for (i in 2:numRecord)
{
    step <- stepFromPrice(dataClose[i])
    previousStep <- stepFromPrice(dataClose[i-1])
    dataStepChange[i] <- step - previousStep
}
dataStepChange <- as.matrix(dataStepChange)
dataStepChange.train <- dataStepChange[1:(length(dataStepChange) - numTest)]
dataStepChange.test <- dataStepChange[-1:-(length(dataStepChange) - numTest)]

# init variable to store direction
dataDirection <- rep(NA, numRecord)
# calculate direction
for (i in 2:numRecord)
{
    if (dataChange[i] < 0) {
        dataDirection[i] <- -1
    } else {
        dataDirection[i] <- 1
    }
}
dataDirection <- as.matrix(dataDirection)
dataDirection.train <- dataDirection[1:(length(dataDirection) - numTest)]
dataDirection.test <- dataDirection[-1:-(length(dataDirection) - numTest)]

# init variable to store ema change
dataEMA10Change <- rep(NA, numRecord)
dataEMA60Change <- rep(NA, numRecord)
dataEMA250Change <- rep(NA, numRecord)
# calculate ema change
for (i in 2:numRecord)
{
    dataEMA10Change[i] <- dataEMA10[i] - dataEMA10[i-1]
    dataEMA60Change[i] <- dataEMA60[i] - dataEMA60[i-1]
    dataEMA250Change[i] <- dataEMA250[i] - dataEMA250[i-1]
}

# split change, stepChange into plus, minus data
dataChange.plus <- dataChange[which(dataDirection == 1)]
dataChange.plus.train <- dataChange.train[which(dataDirection.train == 1)]
dataChange.plus.test <- dataChange.test[which(dataDirection.test == 1)]

dataChange.minus <- dataChange[which(dataDirection == -1)]
dataChange.minus.train <- dataChange.train[which(dataDirection.train == -1)]
dataChange.minus.test <- dataChange.test[which(dataDirection.test == -1)]

dataStepChange.plus <- dataStepChange[which(dataDirection == 1)]
dataStepChange.plus.train <- dataStepChange.train[which(dataDirection.train == 1)]
dataStepChange.plus.test <- dataStepChange.test[which(dataDirection.test == 1)]

dataStepChange.minus <- dataStepChange[which(dataDirection == -1)]
dataStepChange.minus.train <- dataStepChange.train[which(dataDirection.train == -1)]
dataStepChange.minus.test <- dataStepChange.test[which(dataDirection.test == -1)]

dataVol.plus <- dataVol[which(dataDirection == 1)]
dataVol.plus.train <- dataVol.train[which(dataDirection.train == 1)]
dataVol.plus.test <- dataVol.test[which(dataDirection.test == 1)]

dataVol.minus <- dataVol[which(dataDirection == -1)]
dataVol.minus.train <- dataVol.train[which(dataDirection.train == -1)]
dataVol.minus.test <- dataVol.test[which(dataDirection.test == -1)]

###### Prepare Train Data ######
# for direction prediction
# set of data: (current, ema10, ema60, ema250, vol, buyVol, sellVol, next)
# current, ema10, ema60, ema250, vol, buyVol, sellVol: remove last row
# next: remove first row
setDirection <- as.data.frame(cbind(dataDirection[-numRecord], dataEMA10Change[-numRecord], dataEMA60Change[-
    numRecord], dataEMA250Change[-numRecord], dataVol[-numRecord], dataBuyVol[-numRecord], dataSellVol[-numRecord
    ], dataDirection[-1]))
colnames(setDirection)[1] <- "Direction"
colnames(setDirection)[2] <- "EMA10Change"
colnames(setDirection)[3] <- "EMA60Change"
colnames(setDirection)[4] <- "EMA250Change"
colnames(setDirection)[5] <- "Vol"
colnames(setDirection)[6] <- "BuyVol"
colnames(setDirection)[7] <- "SellVol"
colnames(setDirection)[8] <- "PredictedDirection"

# split into 2 variable: train, test
setDirection.train <- as.data.frame(setDirection[1:(length(setDirection[,1]) - numTest),])
setDirection.test <- as.data.frame(setDirection[-1:-(length(setDirection[,1]) - numTest),])

# for price prediction
# set of data close: (current_close, vol, next_close)
# current_close, vol: remove last row
# next_close: remove first row
setChange.plus.train <- as.data.frame(cbind(dataChange.plus.train[-length(dataChange.plus.train)], dataVol.plus.
    train[-length(dataVol.plus.train)], dataChange.plus.train[-1]))
colnames(setChange.plus.train)[1] <- "Change.Plus"
colnames(setChange.plus.train)[2] <- "Vol"
colnames(setChange.plus.train)[3] <- "PredictedChange.Plus"
setChange.minus.train <- as.data.frame(cbind(dataChange.minus.train[-length(dataChange.minus.train)], dataVol.minus
    .train[-length(dataVol.minus.train)], dataChange.minus.train[-1]))
colnames(setChange.minus.train)[1] <- "Change.Minus"
```

```
colnames(setChange.minus.train)[2] <- "Vol"
colnames(setChange.minus.train)[3] <- "PredictedChange.Minus"

setStepChange.plus.train <- as.data.frame(cbind(dataStepChange.plus.train[-length(dataStepChange.plus.train)],
    dataVol.plus.train[-length(dataVol.plus.train)], dataStepChange.plus.train[-1]))
colnames(setStepChange.plus.train)[1] <- "StepChange.Plus"
colnames(setStepChange.plus.train)[2] <- "Vol"
colnames(setStepChange.plus.train)[3] <- "PredictedStepChange.Plus"
setStepChange.minus.train <- as.data.frame(cbind(dataStepChange.minus.train[-length(dataStepChange.minus.train)],
    dataVol.minus.train[-length(dataVol.minus.train)], dataStepChange.minus.train[-1]))
colnames(setStepChange.minus.train)[1] <- "StepChange.Minus"
colnames(setStepChange.minus.train)[2] <- "Vol"
colnames(setStepChange.minus.train)[3] <- "PredictedStepChange.Minus"

##### Train SVM #####
# load library
library(kernlab)
# create svm model with model xxx~., i.e., xxx column is product of the others columns
# direction prediction
svr.direction.model <- ksvm(PredictedDirection~., data = setDirection.train, type = svmType_direction, kernel =
    kernelType_direction)
# price prediction
svr.change.plus.model <- ksvm(PredictedChange.Plus~., data = setChange.plus.train, type = svmType, kernel =
    kernelType)
svr.change.minus.model <- ksvm(PredictedChange.Minus~., data = setChange.minus.train, type = svmType, kernel =
    kernelType)
svr.stepChange.plus.model <- ksvm(PredictedStepChange.Plus~., data = setStepChange.plus.train, type = svmType,
    kernel = kernelType)
svr.stepChange.minus.model <- ksvm(PredictedStepChange.Minus~., data = setStepChange.minus.train, type = svmType,
    kernel = kernelType)

##### Predict 1 ahead #####
# direction prediction
# direction
svr.direction.predict <- predict(svr.direction.model, setDirection.test)[,1]
dataDirection.predict <- svr.direction.predict
for (i in 1:numTest) {
    if (dataDirection.predict[i] < 0) {
        dataDirection.predict[i] <- -1
    } else {
        dataDirection.predict[i] <- 1
    }
}

# price prediction
# prepare starting point
startPlusIndex <- 0
startMinusIndex <- 0
for (i in 2:length(dataDirection.train)) {
    if (dataDirection.train[i] == 1) {
        startPlusIndex <- startPlusIndex + 1
    } else {
        startMinusIndex <- startMinusIndex + 1
    }
}
# change, stepChange
# create dummy column
svr.change.predict <- rep(NA, numTest)
svr.stepChange.predict <- rep(NA, numTest)
# use as point for prediction (update by predicted data, update from preparing point to the predicted data of pre-
    prediction date)
tempPlusIndex <- startPlusIndex
tempMinusIndex <- startMinusIndex
tempPredictChange <- vector()
tempPredictStepChange <- vector()
for (i in 1:numTest) {
    if (i > 1) {
        if (dataDirection.test[i-1] == 1) {
            tempPlusIndex <- tempPlusIndex + 1
        } else {
            tempMinusIndex <- tempMinusIndex + 1
        }
    }

    tempChangeSet <- rep(NA, 1)
    tempStepChangeSet <- rep(NA, 1)

    if (dataDirection.predict[i] == 1) {
        tempChangeSet <- cbind(dataChange.plus[tempPlusIndex], dataVol.plus[tempPlusIndex])
        tempStepChangeSet <- cbind(dataStepChange.plus[tempPlusIndex], dataVol.plus[tempPlusIndex])

        tempChangeSet <- as.data.frame(tempChangeSet)
        tempStepChangeSet <- as.data.frame(tempStepChangeSet)

        colnames(tempChangeSet)[1] <- "Change.Plus"
        colnames(tempChangeSet)[2] <- "Vol"

        colnames(tempStepChangeSet)[1] <- "StepChange.Plus"
        colnames(tempStepChangeSet)[2] <- "Vol"

        tempPredictChange[i] <- predict(svr.change.plus.model, tempChangeSet)[1]
        tempPredictStepChange[i] <- predict(svr.stepChange.plus.model, tempStepChangeSet)[1]
    } else {
        tempChangeSet <- cbind(dataChange.minus[tempMinusIndex], dataVol.minus[tempMinusIndex])
        tempStepChangeSet <- cbind(dataStepChange.minus[tempMinusIndex], dataVol.minus[tempMinusIndex])
```

```
            tempChangeSet <- as.data.frame(tempChangeSet)
            tempStepChangeSet <- as.data.frame(tempStepChangeSet)

            colnames(tempChangeSet)[1] <- "Change.Minus"
            colnames(tempChangeSet)[2] <- "Vol"

            colnames(tempStepChangeSet)[1] <- "StepChange.Minus"
            colnames(tempStepChangeSet)[2] <- "Vol"

            tempPredictChange[i] <- predict(svr.change.minus.model, tempChangeSet)[1]
            tempPredictStepChange[i] <- predict(svr.stepChange.minus.model, tempStepChangeSet)[1]
        }
    }
    svr.change.predict <- cbind(svr.change.predict, tempPredictChange)
    svr.stepChange.predict <- cbind(svr.stepChange.predict, tempPredictStepChange)
    # remove dummy column
    if (numStep == 1) {
        svr.change.predict <- svr.change.predict[,-1]
        svr.stepChange.predict <- svr.stepChange.predict[,-1]
    } else {
        svr.change.predict <- svr.change.predict[,-1]
        svr.stepChange.predict <- svr.stepChange.predict[,-1]
        svr.change.predict <- as.data.frame(svr.change.predict)
        svr.stepChange.predict <- as.data.frame(svr.stepChange.predict)
    }

    ###### Calculate Error and Print output ######
    postProcessPredictionCombine(dataClose, dataChange, dataStepChange, dataDirection, dataDirection.predict, svr.
        change.predict, svr.stepChange.predict, numTest, numStep, outputDir)

    ###### ===== End ===== ######
```

APPENDIX B PREDICTION RESULTS

In this appendix, all result of our experiment is presented.

Price Prediction Result

PredictPrice1 : Time Series

Table 1 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	35.401600949699805	35.434718597763	30.1478798350759	35.3227582920428	35.4589338417676 96	39.7758525811485	35.3618211587044	35.4244476136938	30.4365471873679
AOT	31.220470992528597	32.140636101604805	31.5552487294832	31.2643895454916	32.3094075002785 96	31.7122540671309 03	30.7869422231038 04	32.2669012089437	31.1042932784334
BA	3.03447215550718	3.19633240656476	7.5917677746791	3.14455186615381	3.1342857270349302	7.0562909065015	3.05946925893115	3.1585264356394602	7.6352899847953095
BANPU	52.260590590513104	53.511347804231	13.4543222015549	52.0900832636 18304	52.6884992104947	15.6385424285213	51.7417518991552	53.5934536463641	13.4689182750837
BBL	12.53151147048 03	13.911025034440 2	15.0500525820412 02	12.879083109363801	14.115011537034	13.5631398550137	12.8105653977273	13.7409771687722	14.2859845338635
BCP	5.95543032782214	6.63712338561108	9.5864356794925	6.1875646269560995	6.7476625153099405	7.9467201103882905	5.9531720458054 59	6.747763754358 08	9.6775743552053 6
BDMS	20.2142430921444	20.7181702184355	24.619248879541	20.4294500053712	20.6645367635359	24.8182938209984	20.0878928370046	20.8386468914945	24.0034522360619
BEC	58.2740482962094	62.4060575797054	5.6067256048081 09	57.7209561996482	59.5023612775196	7.9046708963452 7	57.4604790125426	61.5104782084253	5.6581074521876 9
BH	23.0415786429697	24.4290699862929	27.2812467586657 02	24.0034936389483	25.8134482308161	27.029346811529	23.4136381715503	24.6643088576038	26.7758639786985
BLA	20.9516223110226	21.0515247705884	13.7157663619787 02	20.6949748010478	21.000147706619	8.6853186264336 1	21.0021131754619	21.0004909699148	14.4698628456331
BTS	4.94607186153009	5.12377729667497	6.065632592786 93	5.0374313972055305	5.31147189484108	6.6472308255784	4.8755000818532 79	5.3847076449000 7	5.9953963705719 4
CBG	28.1443755813433	28.4828817858597	23.421910676759598	28.5360796317787 97	28.3276040440388	23.3760811937932	28.2598676235727	28.2629810823768 96	23.3572365782905
CENTEL	15.362128446870098	15.2426222077743	10.9462262652522	14.8091794930609	15.230321763532	17.1553905142442	15.4544866510554 99	15.0742415391408	9.6738578713982 1
CK	9.86094771942183	10.9769729833305	18.6243917887636	9.928495252161 09	10.8619021371940 1	16.3210565016836	9.4871386189854 21	11.0007297497396	17.3613563391688
CPALL	7.4476312831335	7.9115158943926	16.4445571994883	7.7890055128671005	7.747267184566 22	7.8433296728622 11	7.4494447246001405	7.8515918465622905	16.1208446677134
CPF	16.768865822543898	20.5392269398663	20.7022047132166 98	18.643656971805 8	19.176833365596 6	25.2047438755143 03	19.5614507994913	19.9965408991011	19.9597565234218 02
CPN	10.795358569211599	10.8530591550638	11.8310758687961	10.8044981321318	10.8522937234226 99	17.1588128203605	10.6949346755997	10.8488836936885 01	11.6725359670682
DELTA	7.468698244733581	8.4254927414566	17.6872182893772	8.526078751556 4	7.9169034993883 1	20.1057969822327	7.3155539140028 11	8.5975667230171 9	17.7008852716586
DTAC	150.96800574648398	158.4505338866602	25.554467621729	155.3256824335621	161.1131850628833	24.9894165254952	155.0722127385709 8	159.7494426175 94	27.1036001047810 02
EGCO	12.8654343910434	12.4843820647069	20.0596149733793	12.6525538528055	12.4899383110888	24.1168112492961	12.7842659821199 99	12.5287138851994	19.3967903888865
GLOW	6.201377611346 89	5.0389561695423 3	10.3357972868889	5.5773242136384305	5.0446707799644 7	10.1128759028849	5.6808949228321 29	5.0432188642004 8	9.4933889005373 6
GPSC	7.365188877131 88	9.2661723777683 9	5.3770862945853 9	8.3323063205895	6.3329983647916 6	3.9500456585121304	4.360487406823 66	8.3799426753676 89	4.7625163165822 1
HMPRO	13.6294779632511	13.1601669911944	39.993801152056	13.3230682085 76	13.180594149256	30.1456565970305 02	13.6635845786490 99	13.2087105364523 01	38.3951430696941
INTUCH	30.8766051563206	31.1356397212640 03	30.1461696168926	30.9530603727118	30.9731780999161 97	32.7846794058193	31.0994656458247 97	31.0935534290694 98	30.2438649613467 02
IRPC	19.152986507588	19.9524739544764	9.661420810424	18.1142765577052	19.9867610351986	10.9222240670341	19.5144185857509	19.9693276085739	9.6816898139058
IVL	14.8071861679241	14.4850161346571 99	16.5159227985381	14.4388461007035	14.5190135345603	14.0361732457234 98	14.8605409850961 99	14.5356754123434	16.1923708070958 98
KBANK	17.5673667530065	19.1769924286805	32.0619224231949	17.3493302508412	17.3270014392135	26.7693560460303 03	16.3876913442048	19.6047083268603	30.3194047007787
KCE	37.8226062989963	42.6708878809278	25.7888860772818	39.7395624119595	42.3735429096611	22.816145445602	40.5947304203881 96	42.7757783760593	25.5291773026220 03
KTB	13.8409958706627	15.8422954235573 01	36.1639105060205	14.8339160739297 02	16.7458026604136	32.5955975661873	14.4826894089303	15.5151479494607 01	34.315957776803
LH	5.57535200465845	6.3899251032793	23.1803123175489	6.354758294574871	6.77102534110395	21.0792504259782	5.935132357618 62	6.4528642199369 9	21.9242495292987
MINT	22.2790402879123	22.7931737773832	41.1364946030278	22.4414896274208	22.7923640190442	36.4328101519146	22.1104310367225	22.7930191142868	40.1056818904857
MTLS	4.38278977199381	4.3882123007765 11	3.1804794796861997	4.50023519658472	4.4326751395142	2.6424273513462	4.3361842147772 1	4.4047794579030 7	3.1925860324359 3
PS	5.16174972176841	5.4451238254979595	49.594860539986804	4.2381765103209 9	5.6736174715881 1	27.4842251252192 04	5.1420892739076 2	5.5277617242232 2	47.354683311886
PTT	16.7192954930225	17.0747085773256	25.3590644604395	15.8354508517033	16.5345030989288	28.2920771362586 98	16.0834116668095	17.1183411806 4	24.2382467833181
PTTEP	73.776128586303	84.2387206417968	30.87629611534	78.6723994462553	84.9463284164875	19.0688594392595	75.8992306330811	87.3207574786622	30.2453992034988 03
PTTGC	13.204021640212199	12.3137901049080 99	14.1701162912190 01	12.2568947976056	12.3118129487385	18.7416702130919	12.3883036180906	12.4446303094274	13.4505016112789
ROBINS	15.390254567064702	15.8108209446489 01	30.4411420557649 98	14.0964467272093	16.6161730048607	37.0716136102907	14.8959354549061 01	16.6525421620519	29.6427084780426
SAWAD	5.74940170477889	5.9028064495395	16.4178163978640 97	5.7580231620102 7	5.9014155901752 5	14.3149165622901 01	5.8491359308144 9	5.8576036367514	15.5366756956572 02
SCB	19.6286628212688	22.6689989301087	27.167424138649 8	18.7098132329042	21.5854717714772	21.0863187239843	19.6118943909 88	23.0230375774390 02	25.7220434098062
SCC	4.85006420397308	4.7232442311354 8	9.0589344449759 8	4.65375095653215	4.6809537471330 6	8.3152291279210 7	4.9096862126387 2	4.6937332666723 7	9.0519049201579 8
TASCO	37.3638137463039	54.3200574683584	67.2731106781299	41.166164933707	57.139232034844	61.4643420560926 05	41.6272365320135	47.5878829159594	65.9610389652857
TCAP	6.859076087062791	7.1004583286706 61	19.9756889518027	6.9509752466388	7.0647621210050 71	19.2999312259162	6.8554056717398 51	7.0924741800051	19.5837320474051
TMB	9.897209190996 48	10.4892493453483	33.748942942283	9.4777445819753 4	10.5330139892931	23.1597195251711 98	9.9972454339342 61	10.1932538614004	31.5290788086688
TOP	17.5681382471287	18.5520082624168	11.2731614269932	17.8363890788793	18.4692901100639 98	10.7111122267479	17.7691024428546	18.5610804071366	11.8563353074461
TPIPL	12.1744899181307	13.2713745119547	14.1101195463217	11.6660667360059	12.5070320587937 99	11.6345011740153	12.0812572481379	13.3376082716345	13.899360121834
TRUE	32.0254842935758	34.087751085204	30.9290156581168	33.158279223024	33.7875285294974	26.2031877726525 98	32.2727033617486	34.1890542146031	29.7416361986782
TTW	4.772333282017 82	5.7304677752663205	9.0967090861656 8	5.1298098853394 19	5.7827929485089395	10.151964945713 8	4.8241914591832895	5.7673863726167705	8.6119379341311 6
TU	9.644651224649051	9.6334905796370 6	21.6749272731638	9.5028142592650 4	9.7041718263660 3	20.2669826295221	9.6658182323777 13	9.7171869861138 8	21.3025475793225
WHA	7.18587259962228	7.4142257593359 7	11.7399496135459	6.7173682594513 2	7.7590490749761 3	10.3351552100564	7.4470887003413795	7.3798426465982 6	11.3443556949468

Table 2 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.7471157669325	1.7222457489899399	1.75889660876105	1.71396174485464	1.71873815292349	1.7142950698118502	1.7697351805322998	1.72337121259497	1.7588229751705602
AOT	1.05768702073291	1.0528605189823201	1.0915139825003801	1.03838512324663	1.04209490914615	1.0588976105730201	1.05148354923009	1.0561411307314699	1.0879963226789902
BA	0.974222110042216	0.951013242113877	0.9265478422880881	0.961063520886362	0.8966300380127461	1.35567234328469	0.9663844938158019	0.93454570710867З	0.917638088484441
BANPU	1.9752542893205298	1.98936519994584	2.02216251450943	1.95153160175658	1.9687391091296198	1.97407319884099	1.99347699100671	1.99029336511854	2.02133538335986
BBL	1.22012293308895	1.22701706165979	1.22654253356103	1.22166197323303	1.23158492414451	1.22410240302036	1.2180153153292	1.2265695623763	1.22651055409095
BCP	1.36741278233826	1.32412925510102	1.39241184938407	1.37423334229314	1.3302460329024899	1.30977909340276	1.36293346395736	1.32257737527012	1.39297014109885
BDMS	0.873030085692895	0.8796448059183121	0.883674919937977	0.908740798802507	0.902526586406629	0.8831395613826809	0.875417644691385	0.8743260697554539	0.8835290149874809
BEC	1.7786338836805	1.68683061823117	1.6866947210041698	1.73120136309823	1.67393109000455	1.6782636757563598	1.7721757712878898	1.67927551531091	1.6852916159861902
BH	1.40218324334679	1.40312759737485	1.39253065905995	1.44774903184191	1.45911722179396	1.42159912783984	1.40196482052513	1.40218783946335	1.39285955811169
BLA	1.61324417460607	1.61311474999512	1.6156854553439097	1.61535715802344	1.62557808080556	1.6562921874272398	1.61302781675351	1.6162765807274202	1.6158618488910401
BTS	0.944764954916384	0.934949298667798	0.938374789528598	0.950490741391532	0.94103006068583	0.9836365690281059	0.937050171984049	0.9343815453184631	0.938210403121603
CBG	1.96873524110824	2.0019325525296	2.07928070343354	2.05159708486379	2.04000503195704	1.9622995645911099	1.96791972527698	2.01394535123935	2.0783091383135996
CENTEL	1.53499144517257	1.52251728431345	1.53805292643456	1.5333588473016901	1.5403234121903	1.56120310139785	1.5322850900345801	1.52584605227947	1.53665431291177
CK	1.35295082974597	1.35900160505462	1.37761390795811	1.35903713613784	1.36166607448651	1.34180415933539	1.35448941639473	1.36456657278818	1.37587062555614
CPALL	1.3271494094690601	1.32234572120431	1.36017279945295	1.3443527605994199	1.32702059581126	1.31239607914223	1.32391108217893	1.32398889148639	1.356500186372
CPF	1.9897994431324801	1.99242109097543	1.98442605882325	1.98572395662216	1.9787321590294098	1.99574261663697	1.9911350591696	1.99061452756112	1.9858304063036898
CPN	1.23642961872198	1.21723673237411	1.19909894953431	1.19396375578243	1.20396582844023	1.20203158381503	1.23164542225024	1.21681782707521	1.19687756117361
DELTA	1.82899598864853	1.7690681409568199	1.80353359702127	1.7477183691200702	1.73641303315522	1.89558553085186	1.8003174493318401	1.7318256964324	1.79976991294817
DTAC	3.20110688010741	3.10471566735069	3.1461280142484402	3.32464739610882	3.14141897744096	3.12650331081733	3.22406484631148	3.10999114662946	3.1374578174826904
EGCO	0.902506427522818	0.8991448549818939	0.947321687702667	0.9160087625697709	0.9002509266591321	0.9511617023454729	0.903447735679897	0.898735304590919	0.94574401842587
GLOW	1.33483810991503	1.27784293842702	1.30256293951474	1.26907928631893	1.26403250637359	1.26764704372542	1.33474106494085	1.2933312919786701	1.30005838936994
GPSC	1.57294343740594	1.51777003903488	1.6438741332441598	1.69216262549164	1.72598640602607	1.81634898465265	1.5875707045995497	1.56325290291368	1.6439541054089302
HMPRO	1.47311370342847	1.47620446088914	1.47249177274679	1.46073163917257	1.45814173895253	1.45489307256674	1.46908468495353	1.4749312771071599	1.47167928809178
INTUCH	1.6363997827739798	1.6011030625951	1.58490426133798	1.5986686507305201	1.59388799523829	1.58647110072575	1.6317014321364998	1.5982149139792599	1.5845948328745099
IRPC	1.29734295618897	1.29102363547397	1.3348221920045	1.3035693267937198	1.30480315549689	1.28392289674451	1.29634534239181	1.29067702357226	1.33207773479281
IVL	2.16644859649977	2.09777106985167	2.10776512214213	2.1013267720677598	2.08094472565634	2.1072437399663397	2.1610377649537003	2.10560090172344	2.10605124942168
KBANK	1.4632266942777399	1.38716397555893	1.42368561599446	1.36339556915937	1.37233182413519	1.38330869950099	1.46600248817309	1.38943883895567	1.42289569633995
KCE	1.47555586641356	1.46580310902005	1.46332768906733	1.45667077022479	1.45644931999628	1.45649646890001	1.47701428223047	1.47983950567405	1.4633904895637702
KTB	1.17492354905927	1.16600331208732	1.17197666950415	1.17344313432004	1.16248950125652	1.17169263683787	1.18863283613834	1.16563535239884	1.17281267572147
LH	1.21641452242945	1.20138340564755	1.26966130650037	1.22487344106401	1.2196547043936201	1.2032960329870102	1.21526381534436	1.20143648954692	1.26612418247049
MINT	1.59549691553439	1.59927174192199	1.6141689688752299	1.6019344023017	1.6082981857978	1.64582207108636	1.5935469450040403	1.59714895088417	1.6138719336023999
MTLS	0.91546903165386	0.9089298251573331	0.8884248921056581	0.914991759066568	0.8980539001926391	1.23173558521228	0.897906714781766	0.9050988943186009	0.8887300848413431
PS	1.23113852907265	1.20617714650401	1.2882166863146498	1.22067861311568	1.19723157523468	1.24728328652721	1.22244174520939	1.2093036343622199	1.28182856157774
PTT	1.8591316551184198	1.86159539266135	1.8451789664201301	1.860451436719	1.81928235805407	1.84076722324173	1.8528104953284799	1.8589042713661401	1.84458886512791
PTTEP	2.24767234040692	2.23991127863265	2.23511360083042	2.18308379188173	2.19212890155403	2.1869581613083398	2.2406848530202703	2.2378621146281903	2.23636379757232
PTTGC	1.7767779042223701	1.7783748834190898	1.79563014968623	1.7742925721434901	1.78548845920185	1.81956446361498	1.7690842053342801	1.78001371907585	1.7952095455600998
ROBINS	1.45950707575563	1.45323490283201	1.47083941859918	1.44814370458516	1.4384146390847599	1.44113372294664	1.45642430214197	1.45218867807649	1.47207449492774
SAWAD	1.41724276884949	1.3722964334697	1.4007242455887101	1.41783203491024	1.40923716156777	1.44725198924495	1.41713322027849	1.36774989236776	1.3998203810663
SCB	1.46425005496825	1.4412859920447	1.60750617094298	1.41850721158343	1.41608288800382	1.40802806382667	1.46292374583351	1.4399018068676999	1.60530632943926
SCC	1.10550600172002	1.08601034898279	1.14541791794291	1.0751498658479999	1.08130982393278	1.22962207700966	1.10674216051397	1.08570097465469	1.14509971780233
TASCO	2.4211476788978703	2.35444362122221	2.73320938050227	2.46354449456945	2.44061148130793	2.5775164891437	2.45926673575397	2.33751230495543	2.72137524510087
TCAP	1.14884340814084	1.1468249433987898	1.17991267276986	1.12949800810687	1.13327606913128	1.1261012553813101	1.15025082831697	1.1514313922241899	1.1785035912131399
TMB	1.38560217269129	1.3935498854228698	1.3955321895867001	1.3885471077262301	1.42366067608815	1.44121089125193	1.38949485645759	1.3940365559518	1.39117229532014
TOP	1.61649227277756	1.53605893520914	1.50783610587186	1.50710942189068	1.50365973683384	1.49986039137049	1.5878670188749802	1.52506024624321	1.50544365299571
TPIPL	1.62213226286065	1.61049507099525	1.62067712141991	1.63688169188617	1.68986174327764	1.7176655593762298	1.6221617063797	1.61405635832848	1.62018088616484
TRUE	1.92079173462512	1.94476949157315	2.05038822818125	1.95722419423902	2.0009339099745	1.97125421679237	1.9172953127616101	1.9436783465385399	2.0445334841503002
TTW	0.79330643983789	0.803215526284277	0.795605622829143	0.785090155634084	0.791201095164815	1.17973009361872	0.791569816219038	0.804272486796991	0.7985957531673961
TU	1.43348868495613	1.36006187988056	1.6095470604152302	1.3607622215103599	1.3430103381662502	1.33148878995029	1.4252056305254102	1.3599088756429	1.5942322389226
WHA	1.55573102158225	1.55613809175733	1.55236536197104	1.57831924882487	1.5817633123602097	1.55916811000458	1.5515512604824	1.54294420453646	1.55285753711673

Table 3 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.74611289976053	1.72079655681521	1.7316986723819001	1.7144877495295	1.72442814712321	1.71371032608688	1.74526106833061	1.72366981889602	1.73139816510763
AOT	1.05356087408046	1.0634785684006	1.11662501661775	1.0590515148174	1.06875917963944	1.14562602142837	1.06318352431306	1.06415718581255	1.10583443366999
BA	0.9679644982070921	0.9420217996553921	0.921965087269766	0.9278068173156709	0.920957832418994	0.933647924468977	0.956840617460499	0.9368092624401541	0.9174214150854111
BANPU	1.97673580581148	1.99127681607224	2.02154919594506	1.94332650269163	1.96887410303529	1.97378279475285	1.99560469385327	1.9925242848690399	2.0219031656916098
BBL	1.21792076982571	1.22667308650124	1.22639529901356	1.21961983674879	1.22901881770505	1.22639566382606	1.2197404977036	1.2266150752068	1.22633679193311
BCP	1.3792108058481098	1.32588654718511	1.39241184938407	1.3760103479313	1.33346194988643	1.30977909340276	1.36912852645844	1.32520415670078	1.39297014109885
BDMS	0.8802635193194359	0.868949264423472	0.8836749199379761	0.906623758093931	0.90012217228621	0.8831395613826809	0.8781768004303241	0.8733753057266159	0.883529014987482
BEC	1.76716261739366	1.67208355342011	1.67961484656283	1.73091513998671	1.6698717477211502	1.68047381864881	1.7537771811008	1.6776786904027698	1.6789430143867201
BH	1.41534293677675	1.40689866294915	1.40497629906209	1.44844321050385	1.47004165182733	1.44392249538386	1.41676809490071	1.40984617777077	1.40481477357646
BLA	1.6132955334591599	1.61433937676133	1.61556377899191	1.61692758841636	1.62630201478444	1.65675277530861	1.6156323516521	1.6140767453077898	1.6158618488910401
BTS	0.946548220940833	0.9291608364219269	0.9349468923727551	0.954818139667347	0.924763372504444	0.982511866104755	0.93257314733321	0.9293040892653931	0.9344968876878971
CBG	1.9686165302280698	2.01881383316021	2.07928070343354	2.0402509779466	2.0421536368146898	1.9622995645911099	1.9790470914244198	1.99857941298061	2.07830913831361
CENTEL	1.5312076427651902	1.52691314427599	1.53816418761102	1.5470634697348	1.53909207441341	1.56120972375017	1.52974367254138	1.52717014230941	1.53767407844862
CK	1.34819346667864	1.38726362594819	1.4123939196511301	1.34571861879254	1.36745737421536	1.490804310107	1.35584556296117	1.38578185628253	1.4097855280832299
CPALL	1.32868608564236	1.32373132027414	1.36017279945295	1.3458033626972299	1.3236855393103	1.31239607914223	1.32720396365941	1.32379198861286	1.356500186372
CPF	1.98799369634437	1.99160176814864	1.9886929839722098	1.98108203115789	1.97805975397201	2.0021207573143602	1.9978211944797	1.99098673491082	1.98743711458674
CPN	1.24233828244131	1.21537400812971	1.19909894953431	1.19387663348681	1.20324963506332	1.20203158381503	1.23092662115231	1.21553873841498	1.19687756117361
DELTA	1.81067655575304	1.73823122262791	1.80353359702127	1.75706783246043	1.7356803164267802	1.89558553085186	1.8183303978843002	1.73086245841115	1.79976991294817
DTAC	3.20242350600508	3.1018689720601103	3.11366238491211	3.3454563847436503	3.15732367119407	3.08455189699451	3.21029549518468	3.1082083892296697	3.11023392692726
EGCO	0.902593565108172	0.898786040072749	0.9547288102177649	0.9160109318404089	0.900289810572111	0.9511431431425771	0.903110991391347	0.8987886759606999	0.9527705824848159
GLOW	1.33427176031486	1.28128260965944	1.30222635768686	1.27731700325286	1.2642928183944402	1.26768087405008	1.33197030112522	1.28073863427785	1.2999096774010699
GPSC	1.5729289099334898	1.55770079930291	1.64387413324417	1.7462604456475	1.70990455638554	1.81634898465265	1.58792068777646	1.53592468241495	1.64395410540892
HMPRO	1.47165145496963	1.47656776242638	1.46864526415475	1.4615922057063	1.45777213487299	1.45372523603676	1.47224654911137	1.4746891599687502	1.46773771386986
INTUCH	1.6256071505886	1.5988213152050599	1.58490426133798	1.5988114945957799	1.59420178650102	1.58647110072575	1.65605776884638	1.59818285997074	1.5845948328745099
IRPC	1.2969595112653	1.29023434512557	1.39862357055534	1.32022242133792	1.3170469102337001	1.28453699480356	1.29521263010807	1.2899574776740599	1.39265265909979
IVL	2.17100525164376	2.10833977076425	2.18189630550815	2.0815382372644997	2.0858761228594402	2.09892875703503	2.1680806985487298	2.10230269332936	2.17626566380856
KBANK	1.42362278731892	1.3839879508353101	1.36673655691941	1.3590632245652001	1.37244228642009	1.36799787221071	1.42497911637173	1.3866942307155299	1.36837150801671
KCE	1.50059075910456	1.47307368221348	1.4629177020052402	1.45817192042528	1.4581687430302699	1.45788967792646	1.50222025597679	1.4764780115680398	1.46308997023758
KTB	1.16915183782674	1.16492095228239	1.17197666950415	1.17308688146491	1.16240360966321	1.17169263683777	1.1786217913544002	1.16795874523473	1.17281267572147
LH	1.22036914473835	1.20108790947778	1.268837581851	1.22145128051544	1.2130196370914301	1.26411521338056	1.2118456913684499	1.20167450811194	1.2648261288087599
MINT	1.5936801391191802	1.60412644455963	1.6222532465237398	1.61933348052738	1.6333592293852999	1.7121807057039002	1.5919200500375101	1.60561127957991	1.62098206267357
MTLS	0.898588060643475	0.90665677887344	0.888424892105655	0.954867588009513	0.887874022245912	1.23173558521229	0.8997533583174301	0.9078010727120001	0.888730084841341
PS	1.26789514663355	1.24041630608074	1.42262958292657	1.19630373880716	1.23695587727041	1.30636421299303	1.26208227511753	1.24711140412762	1.4015608768278298
PTT	1.85047521423567	1.8679135652210102	1.84390958017846	1.86359755840643	1.8200397683378	1.84013123659378	1.85315292966306	1.8660212130804	1.8433324084315998
PTTEP	2.19743317666669	2.20220759492847	2.25159894531404	2.2278396971275	2.19572119736448	2.2166105573371198	2.21465901537856	2.20174380558525	2.2520782433001902
PTTGC	1.78749079093713	1.7851621064296601	1.79563014968623	1.76873272845261	1.78382214497737	1.81956446361498	1.7696951668185399	1.778834120467	1.7952095455600998
ROBINS	1.45300227783537	1.45384615586406	1.47083941859918	1.45034143564656	1.43652681342496	1.4411337229466499	1.45201519586353	1.45224535096262	1.47207449492774
SAWAD	1.4172520438099099	1.37111747661032	1.40072424558872	1.4067580424450599	1.413217473917	1.44725198924495	1.40604892997686	1.40539191665644	1.3998203810663
SCB	1.44993105622128	1.44137650823874	1.60750617094298	1.41933015835246	1.41353654843146	1.40802806382667	1.4623093494253199	1.44038187508767	1.60530632943927
SCC	1.09437456824229	1.0878505011205999	1.14690743741965	1.07908129023665	1.08156243518122	1.19292155325174	1.09067305639128	1.08805412532812	1.14583775021249
TASCO	2.4923066879358	2.39891387825208	2.8636024127907898	2.44655467901547	2.48747825603808	2.60105123600436	2.4957585362598103	2.40015010964003	2.85453728735258
TCAP	1.15184068380407	1.1486924121961701	1.17991267276986	1.129440240389	1.1330795983165398	1.1261012553813101	1.15024973915181	1.14727774695198	1.1785035912131399
TMB	1.4042300653200799	1.39555941248993	1.39555466765741	1.39410105430216	1.42760889473456	1.4384221336257699	1.40276330811164	1.39656115385414	1.39131190339141
TOP	1.6090466225090299	1.5355562972596999	1.50783610587186	1.50904272379463	1.5035866705452599	1.49986039137048	1.5853342243530002	1.5355628426507502	1.50544365299572
TPIPL	1.6234745202452698	1.6128059067633298	1.64555921632351	1.76446969166532	1.74699243029094	1.65945904690234	1.6119259950880802	1.61086887986649	1.6471725109811102
TRUE	1.9142767398878098	1.9373956215467998	1.98745882059327	1.9451422116641401	1.9554887391752902	1.99917207049784	1.9090255774565998	1.93724461544524	1.9833989517123298
TTW	0.7904520569719881	0.7973726728821301	0.7943804258473199	0.811254713333575	0.7923539147522539	0.7825460393928989	0.78779805041807	0.796662114880875	0.7949726109553759
TU	1.42187226871244	1.3806957591459401	1.6089440726229198	1.37645153893409	1.3410701609806799	1.33148878995029	1.4362455389277	1.36694313237986	1.5942322389226
WHA	1.55548759452424	1.54604130595162	1.55325806487354	1.54232919493341	1.57792325689503	1.5817093168310599	1.5520173290671	1.55970749488416	1.55335797819801

PredictPrice2 : Historical Data

Table 4 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	32.8096149499885	32.6427691735908	34.0479711610097	32.3918171949266695	32.3630037232622	32.548696200848	32.9298442106619 05	32.6639983530358	34.1661930645251
AOT	29.9208738440571	30.4550442226775	22.9468745947122	29.9266446465417	30.653491746575	23.3952751144727	29.8900684901081 03	30.4591925442163	22.6526246848853
BA	1.51622271122926 99	1.41485529896392	1.34944273555541	1.42467248563607	1.46001452520971	1.33432697523191	1.5091153913 5	1.41756427464972	1.34818136206794 01
BANPU	47.6551107816045	43.5315893065285	33.0519607274098	47.4711010644647	45.9669773482801	35.795100506755304	47.167714772066	42.909171223430995	33.0861091160169
BBL	8.093960487736709	8.72007463621782	4.94648800491981	8.591435792016 15	8.01576683784457	5.059583177697919	8.427826336047 78	7.70435531280274	4.91962763569944
BCP	1.98244580118386	1.97085142688877	1.87589006886877	1.9568580245288298	2.01457785336161	1.95487626366209	1.9902213611876	1.98715191963251	1.87561796313952 02
BDMS	17.3554067469987	18.9562158532767	9.42146080484724	17.950335362681898	18.5648176333684	9.80830191015891	17.189844708146403	18.9150442624686	9.23225597488852
BEC	42.7906873198888	43.1365858677158	23.2062713431211	44.4211209258481	41.5098220239117	26.018867633072702	43.9284134251838	45.461152992935	23.9683178031492
BH	2.50128658991230 02	2.33344532670479	2.08525737877177398	3.0327535160297	2.59958411786652004	2.154888499184555	2.50466768832721 03	2.34768550477344	2.08937836981543 02
BLA	22.4930164108549	21.6572134465911	12.361170863332	22.2379656420591	22.451890948737997	13.977340293159399	22.476965407039902	22.3612981489835	12.09278128201 57
BTS	1.44186862167679	1.50375621139906 01	1.24206188640768	1.5860854409621399	1.54410357764035	1.335590903755204	1.46160809974733	1.53652445755933	1.24242509353287
CBG	27.0572880364883	27.1501546197844	27.4587109746052	26.7475648108442	27.0770056099218	27.054684510363604	27.2180613648712	27.2051587735669 97	27.4451115467016
CENTEL	8.854978398430 41	8.261196843880569	5.793683646020 46	9.302962645677 62	9.59341279697322	6.080178774259 82	8.663186410004459	8.441224732224 44	5.71788097664588 1
CK	4.558292943254 91	4.96428912245404	2.75816910507145	4.597859579575782	4.67693482278181	2.764308258097 48	4.538999821721859	5.11155624076212	2.73217721388711
CPALL	3.377039766959 14	3.06692015207787 97	2.34358411404981 97	3.5140372831060396	3.24081078111371	2.431297557446409 7	3.442144406169 45	3.249427156519104	2.34581412360365 03
CPF	5.339531928110 79	5.54749099276129	3.20321035813502	5.584878664955 08	4.89746078009365	3.323783331494649	5.122883556414 36	5.17884678515236	3.21544865183387 96
CPN	11.048654048988 3	10.940693828454 4	10.5405387610441	11.16606903964 76	11.038071498138299	10.888709354469 05	10.949363140773	10.851468643324399	10.3098841853849
DELTA	2.496823819173198	2.69446370653276	2.22242811584404	2.67114098187438	2.75000809797143	2.271162926768 45	2.659698659303481	2.87638747487322	2.22310040313923
DTAC	154.803611323786	155.442349215152	101.160254743781 01	152.9198110488 46	150.888806857045	106.935230315422	156.287148895442	153.271614911033	101.74892467985399
EGCO	9.272412089911 07	9.42452947083043	5.82717525210868	9.167058495417 37	9.42086908277351	5.583159315394591	9.308182539441 23	9.55909713217325	5.81103090346083
GLOW	1.732715528620699	1.76030887542392 02	1.68344338701685	1.8671653668692199	1.79487249738112	1.796430356491 89	1.742237009276680 1	1.76413922995584	1.68365585709358
GPSC	6.435473441869729	6.66269384768339	5.08646978620031	6.801631606580689	5.93406232972677	4.473772690022625	7.876390456572 4605	5.22531115852214	5.51324148486756
HMPRO	6.197135989122 87	6.04480387314666	4.53820029685838	6.171148353499 63	6.253650562400789	4.428651828979 95	6.147598786933 23	5.9657380008446	4.54281592083961
INTUCH	29.791245557879297	29.214057335316102	29.698969665194003	29.727490097541697	29.1655719347959	29.084444240684 1	29.861761769706202	29.851528379734 1	29.800863087115896
IRPC	12.089370264379 9	11.8495601263272	6.3990187234472 41	12.1352211161014	12.5598585285003	6.761360339112681	12.219684311123 4	12.234502396156	6.50609914317916
IVL	10.4045282297242	9.92288251138886	8.75896916599894	10.3768672415385	10.0764650000621	8.953948892170 89	10.390537365222 2	9.81975453463725	8.66285711195782
KBANK	5.375603236490 02	6.79441138913781	2.95362856912965	6.1594590240277 6	6.2468779605929505	3.134336045833 74	5.311346717126 49	5.833643842808 1	2.95145542014278
KCE	36.184065440268	37.2634710741157	31.636846236564896	36.5849708652446	37.1500003674388	32.068210933472 6	36.097157255694 2	37.428044575276 6	31.4699476950623
KTB	2.5817038617703	3.06418063177552 02	1.5659096492487001	2.4967249606889 2	3.36247293364421 03	1.632988160827 36	2.635027907673 03	5.07068209619436	1.56410173934475
LH	1.746774288909740 2	1.63984931102 71	1.720634148042	1.735029977487 77	1.66885439947323	1.655512695646 45	1.725985047411 82	1.63476128040488 01	1.71896746616768
MINT	22.16757546956	22.387340388122 4	14.421098729027301	22.199455397971498	22.3752037940804	14.940330512552 22	21.895805408211 6	22.338980112072 67	14.024911493999001
MTLS	3.931734559365705	3.44414822009955	3.37378151548044 03	3.9890307380484296	3.32634261202154	3.538101319189 83	4.031636856592 64	3.54309510836857 95	3.36340923302817 97
PS	1.713907652353 63	1.94313509678831	1.70066760314106 01	1.789118561104 51	1.95296503579987	1.728522906402 689	1.724284269544 198	1.94207980166070 98	1.69912329448415 99
PTT	8.943785329193 53	8.10912194893363	4.87153408487843	8.904449433556 42	8.6546584925234	5.490798001210 57	9.112275552760 67	8.71258202886555	4.88037261160889
PTTEP	37.551816279941 406	43.8780990885667	11.5994319468988	38.239944972333504	48.4521542497732	11.599301162767 8	38.883259857101 7	44.940674542726 5	11.8620078147084
PTTGC	5.769743320816 56	5.89856686366717 9	3.78454456026492	6.03828647432357	5.78156662925691	4.011101728671 1	5.778193536198430 5	5.41856750910528	3.79242700558622 97
ROBINS	6.894376633911 615	6.77294154061 24	4.67781474868596	6.811094991383	6.80435671782183	5.106148567171729	6.83513319236 2	6.73096984261704 1	4.69461307348781
SAWAD	2.300939139372 03	2.31361945225487	2.10952415799171	3.092038246907 2	2.31497268797549	2.142945523228 08	2.364595278726 94	2.65176769153655	2.11144172576577 03
SCB	13.004743847605802	13.323597235113699	6.87183235323069	13.5700835571909	13.505549784831	6.890183210498	13.163123884261902	12.685639988606201	6.88462409420854
SCC	1.535785864294269 9	1.59104390598941	1.47154996482461	1.6758924574584302	1.58070905586410 02	1.498236892866399 8	1.588932375694	1.652181614784 69	1.47153046194564
TASCO	19.917408067015	42.0982798709642	3.70783154197585097	25.5313339813885	43.4740630769081 95	3.547603106578 53	21.169804481223	44.411539456173 3	3.69201902156 54
TCAP	3.027085004453 23	2.77180247348031	2.22448358628481	3.0923590768175	2.93853526846592	2.511379258999 28	3.016287962229 55	2.88098120652366	2.22836456569833 97
TMB	2.264138363624 22	2.28255748748051 97	1.73177655435213	2.472540530361 87	2.25015154677109	1.754965305073 42	2.691353331124 306	2.63018775195607	1.73150320065474
TOP	19.000116216336 3	19.259334096898 7	17.111183700378	19.041966716791 9	19.311683355269903	18.761414720634 8	19.993383362653 5	19.215340253737 98	17.042200437337
TPIPL	3.103414488765 959	3.87755020338148	2.31475530382418	3.3154837084082995	3.53285770575812	2.366259771029110 2	3.147277612812 25	4.55085152673516	2.32005772510501
TRUE	7.886759382076 14	10.718998524644 8	3.04358695818343	9.462377975196 99	10.607157196798 1	3.154480812313 94	8.793476734908 53	15.151149360220 6	3.03841493144638 97
TTW	2.006029301009 43	2.04733136144672	1.09069326696044 99	2.704150809044597	1.91144091537871	1.113329921431 31	1.846210886866 29	2.01386337233623	1.09270546123772
TU	1.971262935581 7	1.89270832150978 02	1.69846955654128 02	1.868831927113 46	1.92880072013416	1.707262244890 92	1.933849339350802	1.981572617215969 8	1.69910681610403 01
WHA	2.146441656554 44	2.21606738358994	1.97947143219809 02	2.246104043757	2.2257422076619	2.156712995288 75	2.126042132816 73	2.22483335898752	1.98009369794412

Table 5 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.84645833996785 98	1.80995349601936	1.83418979747191 201	1.741914886315 82	1.750900428096	1.764911561209499 9	1.845031724521 2602	1.80709538752216	1.83372632263655 99
AOT	1.097455084620699	1.05115367194771	1.07328183750526	1.0452381897183	1.018838486919059 9	1.021041754082	1.094653310148 53	1.05088926777983	1.07327581531368
BA	0.893105227949892 1	0.90049701617498 3	0.92554174358338 79	0.9798408112261171	0.923690784101458	0.964624648039046 9	0.892664331533148	0.90158631840219 89	0.92421472939398569
BANPU	2.04179450408438	2.04308308331747	2.05320579839527	1.997163304457 36	1.99863971043641	1.997880128733 25	2.06850319865024	2.03358918366716	2.05347314676642
BBL	1.22012647450159	1.21235400849770 1	1.24877587370094	1.228083656528 22	1.21654720033797	1.247765752336 66	1.232407359203 67	1.21288731488605	1.24873247327697
BCP	1.38895840629551	1.35748063498287 99	1.39210129933494	1.381239365759	1.34919906302817	1.380084075617559 9	1.390176154579 4	1.35799685150114	1.39229481566405
BDMS	0.934401361491410 1	0.892013233131327	0.90058115423672 19	0.924525353093932	0.895200376718123	0.894952611308329	0.934135683090211	0.893623071795619	0.90051510511299 8
BEC	1.873891112609 78	1.79864993960832	1.77418801716829 8	1.887362952196949 7	1.82986740077851	1.886173540882429 8	1.869025800863639 8	1.801361091998	1.77424740176886
BH	1.5173168726729	1.45239774817029	1.50336894683969	1.501231419674380 1	1.46894386910363	1.533876466954 01	1.521611813801465	1.44715012495683	1.50319164389001
BLA	1.704554728093198	1.66961262924494	1.74148182133123602	1.713587392440201	1.67425866784896	1.719983624477 98	1.705104641105 55	1.67164928955927	1.74202696716923 01
BTS	0.971986901593717 9	0.95966094151623 6	0.98168779076035	0.972018010360 39	0.979735712493281 1	0.981307202958868	0.972168941473105 1	0.959384381376154	0.98185157019449 8
CBG	2.028331755201 9	2.03010810628333	2.0767535110931	2.0570051613647	2.08674855058899	2.061091000177 71	2.032954882044 64	2.02856327262732 03	2.07888366934899
CENTEL	1.600375860519 46	1.54335423211179	1.60908081905184 02	1.556601938102 31	1.516462184858449 9	1.566738845733402	1.604033877207 23	1.5478407729579	1.60893414861323
CK	1.435597240871470 1	1.40539178058248	1.43274667223405 98	1.409938222891 27	1.37247420562627	1.404296702894 9	1.435060819654 77	1.40591043133385	1.43243779609135 01
CPALL	1.395550285789 24	1.35333311572832	1.38719045422027	1.385047124830 44	1.36831121957002	1.386311666081115	1.396188950664 68	1.35233646156775	1.38634366352402
CPF	2.058053586749 22	2.04525900110024103	2.06281783388093	2.0951659385437 2	2.13973167411296 04	2.135819542058 98	2.058664682816 684	2.04734339069863	2.06277853208702
CPN	1.227800377693940 1	1.21374799005 72	1.21295264086283	1.225959459434 34	1.20286736296 16	1.234471369130 78	1.229184605098 59	1.21182376084 37	1.21300964641967
DELTA	1.825536340566 12	1.80505126874739 99	1.78446211043764 01	1.846888885566 49	1.82442358623980 02	1.867584762736 97	1.825258073022 86	1.80458384639078	1.78456648020438
DTAC	3.170081372531403	3.17629462131 52	3.13824521477528	3.155522383306 03	3.08294698693058	3.104410178056 3	3.168409323693 63	3.17793408373747	3.14005122726666 03
EGCO	0.959345958523109	0.95078465834123 5	0.94825563529067 4	0.951201998979402 9	0.943896449714823	0.935425182309465 9	0.958795368664387	0.94978810968958 4	0.94810594808151
GLOW	1.395042373718900 2	1.34362794388692 99	1.38123788552268	1.388338983432 75	1.34809605080984 98	1.358879006641 14	1.391633382666 033	1.34286080604994 98	1.38128292006791
GPSC	1.648749211159 61	1.58687372271515	1.69177353681079 98	1.734230052757 97	1.71277981575027	1.761595713497 63	1.700970892204 23	1.58341574990099	1.69088672173253
HMPRO	1.541539272777 11	1.50031582484725 01	1.51120319047507	1.492297444367 42	1.48708032295154	1.501066227472 75	1.546961800625 23	1.49456356729162	1.51056367644184
INTUCH	1.660851766418 91	1.63675878543474	1.67698787693327	1.625239765066 28	1.599165978343	1.617472178945649 9	1.657525151482 393	1.63715220143359	1.67695732715575
IRPC	1.323287748941498	1.31204647428283 02	1.31488566621287	1.342238533696 63	1.32106923399 74	1.339253329768 37	1.322940057637 78	1.31095131036173	1.31475607462628
IVL	2.19195207833 13	2.18466345865 38	2.16090320891167	2.190001153959 9	2.14979356060997 03	2.160968347083249 7	2.191285489922 155	2.18520160238 18	2.16059806216156
KBANK	1.486536005828 81	1.43866789711804	1.51101201288185	1.426452909813 87	1.41286258730843	1.445835619432	1.486112240768 24	1.43892053591697	1.51141786359461 98
KCE	1.473294627355 01	1.46082395638967 99	1.50962797602445	1.453378833108 72	1.45159837671784	1.470325026654 58	1.482383724658 13	1.45890843526563	1.50977344037323
KTB	1.287004096150 07	1.24665552463533	1.23284403648688 198	1.260994815832 96	1.22134640912957	1.217468653379 17	1.293210113707 88	1.24411232147680 1	1.23275838821613
LH	1.265183562078430 2	1.24384009471668 99	1.25174554913459	1.22729102842 01	1.22904717997873	1.234920722285 62	1.264712690786 4201	1.24152397179413	1.25204341025505
MINT	1.655120047768 39	1.63229751353666	1.70015354727166	1.632388056385550 2	1.63547326866	1.670638168580 67	1.654884286187 21	1.63222710935571 01	1.70016289868579
MTLS	0.843963063992795	0.83785645929786 3	0.84856593560631 91	0.898222990972572	0.852883263476759	0.853415421632650 9	0.835183349083967 1	0.83837112482697 2	0.84733016936512 21
PS	1.298236490187 48	1.27876720949578	1.27765591245907 98	1.304719270191 18	1.25675709064249	1.260172325304 41	1.312138469643 42	1.28085605203234	1.27771673145133 4
PTT	1.912343222860 23	1.88270807181232	1.86588381323512	1.872458752176401	1.84308076394722 02	1.862134727690 79	1.933163845939 5	1.88338152976077	1.86524921482779
PTTEP	2.200040202098003	2.19605027026623	2.13951705564508	2.269960006656 37	2.23743873150419 6	2.187008345502 63	2.220279865093 758	2.18945246061534	2.13959407409060 3
PTTGC	1.870141555980251	1.84621787366518 02	1.86961748762306 98	1.882990734140 93	1.85126321585059 99	1.871300564484 54	1.870082497015 701	1.84287347052091 02	1.86961859494652
ROBINS	1.561660595816 28	1.50349873292806	1.56723814177824 02	1.513349288894 58	1.45923046090829	1.506189607671070 1	1.561596140611 3998	1.50636787457503	1.56713230714488
SAWAD	1.535151302115 901	1.43117807919103	1.44955213081703	1.491843104393 9	1.38675784022308	1.429390708587 62	1.521806760155 9	1.40072521904716	1.44766583065983
SCB	1.507951318102 99	1.46083451758527	1.49164955487663	1.468963143911 39	1.41417096666984	1.463199043953 09	1.504683520465 6301	1.46225736884762	1.49200059418767
SCC	1.140484561830199	1.10778100188136 99	1.13129952982161	1.11982497955824	1.10807245051499	1.128675984638 61	1.146263382710 26	1.10637888709225 01	1.13121462051159
TASCO	2.417813253554 03	2.43162647631441	2.60180515472133	2.470387074454 66	2.46921260589669	2.525588541452 06	2.418877085695 48	2.43498509625342	2.59702729987574
TCAP	1.188374338616600 1	1.18074124872855	1.18090009348209	1.184055983614760 1	1.13323231196849	1.137247907069 73	1.188767291184300 1	1.18030087587244	1.18088516158916
TMB	1.439835158801 38	1.40930690374938	1.43777817787545	1.456013600525 01	1.44181990955134	1.438990990065 78	1.440213687297 59	1.40663794074544 01	1.43765719667613
TOP	1.536269693401 14	1.51286273189222 99	1.52472131543985	1.5283734387	1.51252047142953	1.554502110748 54	1.536248116950 23	1.51275291008819	1.52460409330827
TPIPL	1.668883777754 63	1.64497479162244	1.69078220221297	1.769202258432 61	1.7441796141902	1.796597152409 9	1.668994558875 89	1.64206370070544 02	1.69088931948515
TRUE	2.023798068147 97	2.01129055327906	2.02169452438672	2.042730123514129 7	2.06101433355015	2.073768093335260 2	2.021584439631 0002	2.01349082843308	2.02310179186619
TTW	0.823769723160941	0.83493598410869 2	0.82788408910104 31	0.851911223046311	0.842513651991662	0.860591682737665	0.825149149769683	0.83935250623863 4	0.82802998525230 8
TU	1.347710398545 39	1.33702754543395 01	1.38293926471434 01	1.394068232334 78	1.35121363213203	1.375898347483 2	1.356374945452 27	1.33887968225945	1.38283932286505
WHA	1.619185056629 66	1.54064120153137	1.59386337555908 99	1.676911429518980 2	1.66073161355563901	1.637570730737 51	1.616124654322 8098	1.54133941665614	1.59362249671 13

Table 6 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.78217955163455	1.77350708299409	1.76540545691154	1.73870942048928	1.73756143537823	1.74044620295932	1.78538888278234	1.77597622817614	1.76539891493903802
AOT	1.128821542474651	1.09071814987514	1.074597668820626	1.074045239040422	1.07002315913359	1.064861853968444	1.132136759368667	1.091517833453381	1.07421105875719
BA	0.887699288370269	0.871621790242081	0.896940024698441	0.943260666227126	0.897262622652319	0.937268711857831	0.889704010992127	0.871505735631479	0.89873122680679 51
BANPU	2.041001721292225	2.021513135945097	2.042981803177782	1.99128028503441	1.998926908043911	1.999916372991961	2.049839916411970 3	2.033765961826 01	2.043217455631 09
BBL	1.225515053363 21	1.220896946287 41	1.260205785620 53	1.239515757209299	1.233434823108 5	1.256435470483 16	1.224742260197 43	1.221150965820 79	1.25987581880 874
BCP	1.391144202613 11	1.357559183771 33	1.392101299334 94	1.381490672211 89	1.349120677935 4	1.380084075617 57	1.396335487395 26	1.357622070855 28	1.39229481566 405
BDMS	0.932132670141547	0.894441305288237 1	0.900581154236522	0.923501905025755	0.894756022056993 9	0.894952611308 1301	0.934115713411817 1	0.892276650085217	0.90050538171005251
BEC	1.871646975067519 8	1.795522322072160 1	1.768480626774 14	1.861407623661 86	1.815650977903629 8	1.847144412686 99	1.898944950094 54	1.801248171181 46	1.76841892407526 01
BH	1.467747111376 77	1.423242138605 57	1.502926240732319 8	1.543697288742 79	1.505009690237 27	1.552934527078 3101	1.476762993393 92	1.424289519076 09	1.50274415807 681
BLA	1.711925410194610 2	1.672419083302190 1	1.741481821310160 2	1.718110799492 91	1.674668214552699 9	1.711983624477 9	1.703595386447 79	1.668029257429 33	1.74204211584 119
BTS	0.9644291700996	0.958175261312574	0.979149676596129	0.957741724607808 9	0.959460982093553	0.954579409247319	0.963711663084131 9	0.957102088078541	0.97882219297198 09
CBG	2.033126471855 28	2.032758660843 51	2.076753511093 1	2.055041057478 96	2.073094375044939 8	2.061091000177 72	2.032868015334 1097	2.033582837203 355	2.07888366934 899
CENTEL	1.611571927282 63	1.545738391891 53	1.615503335412 29	1.553903426726 46	1.515724669060 51	1.566131383660 2901	1.611258476212 884	1.546900584943 43	1.61551184982 356
CK	1.459145408995 12	1.426917870547 78	1.455371068663 74	1.415967314462 43	1.395393455451 1	1.454612236997 01	1.457410514034 22	1.431197630647 56	1.45508515402 217
CPALL	1.389875981308870 1	1.352665459248 05	1.387190454220 27	1.385923440456 18	1.369458602020 82	1.386310666811 15	1.396715860980 27	1.352205603864 09	1.38634366352 402
CPF	2.050195957687296	2.020817508443 94	2.026423056160 31	2.019131463832 45	2.097201409656 66	2.128496869371 19	2.049365132246 57	2.019380949464 0202	2.02596815542426 97
CPN	1.228094357421 22	1.213839966635 3	1.212952640862 83	1.226251237467 54	1.204431497687 8001	1.234471369130 78	1.223938924346 99	1.211691467927 69	1.21300964641 967
DELTA	1.825631892919 34	1.804699776923 2398	1.784462110437 6401	1.849339819827 43	1.822874593195 16	1.867584762739 67	1.825662145968 75	1.804656123086 7901	1.78456648020 438
DTAC	3.154918111401 01	3.147680022358 29	3.150247234054 5102	3.116923912175 44	3.077945420684 09	3.125621645823 3697	3.152866188340 04	3.144067033724 2205	3.14991225109 63
EGCO	0.958806373854 494	0.949274612563 1829	0.947746658678 585	0.950877899503 434	0.943604553076 4201	0.935677287300 6189	0.959102709286 599	0.948278177052 8199	0.94759179699 84889
GLOW	1.389762280372 08	1.337862457423 86	1.375778728072 78	1.387977042467 94	1.347569696207 12	1.360095838982 05	1.387161961269 4	1.338323292315 672	1.37599070523 105
GPSC	1.703167349727 202	1.592884245926 002	1.691773536810 81	1.731490511194298	1.721226683304 8302	1.761595713497 63	1.670846494088 2201	1.596753944236 72	1.69088672173 253
HMPRO	1.548142473879 84	1.505384321381 71	1.513549714387 4699	1.497349730214 06	1.485931739361 65	1.505831360805 13	1.543868777464 02	1.504186157318 43	1.51316409719 114
INTUCH	1.663136965199029 8	1.636820201512 5601	1.676938786933 27	1.624897832291 82	1.600051363706 87	1.617472178945 6499	1.656805503255 9001	1.636608317362 38	1.67695732715 575
IRPC	1.307298708281 92	1.301136591751 42	1.310516912119 5102	1.344508572343 0898	1.334381092905 8901	1.312394460582 2	1.305794807856 6	1.300491440073 9	1.31036448643 09
IVL	2.254032579516 68	2.183728007253 68	2.208597469572 35	2.203847586058 9197	2.167870364140 1302	2.221533587035 87	2.253527280322 2	2.187425284109 42	2.20847037675 511
KBANK	1.522276665771 36	1.470096377828 53	1.510095453119 4	1.436461126202 22	1.428912754549 9	1.470359878620 04	1.522309376764 44	1.464590084377 52	1.51030672755 015
KCE	1.481034959240 2599	1.465710956136 09	1.518098436529 86	1.460328987971 37	1.463887848560 84	1.495051853849 92	1.471468375079 62	1.464357347113 03	1.51804894209 248
KTB	1.285649733448 8899	1.246849962781 1	1.232790441242 57	1.255186810680 3599	1.219044355685 8501	1.217468653379 17	1.280556393276 3502	1.243279742744 09	1.23266538648 107
LH	1.271595765832 16	1.248876126656 93	1.264324226287 64	1.278568882830 601	1.246509091646 58	1.282185655878 15	1.272418831459 41	1.250125334449 34	1.26442743209 616
MINT	1.691448763103 49	1.648218894665 799	1.683258446082 38	1.669406314617 58	1.647524712442 63	1.676755372100 0001	1.689795607045 8	1.646331046747 36	1.68306394100 547
MTLS	0.851444594726024	0.840969979031 2799	0.848565935606 421	0.900451747016 46	0.848543755156 9559	0.853415421623 653	0.840405380928 348	0.839811636788 376	0.84755751394 7069
PS	1.312648359508 92	1.316295334150 44	1.262058072522 13	1.322311643782 61	1.283742183034 95	1.304927434207 06	1.315200974422 66	1.310082714709 001	1.26282019925 851
PTT	1.930566793577 71	1.887149187556 8298	1.866489614927 7501	1.874822565429 56	1.839451022329 0201	1.861651522237 94	1.929448363193 5502	1.883242902999 2301	1.86588448142 695
PTTEP	2.260369347530 27	2.197139584707 3802	2.196636743070 78	2.251197984134 36	2.229018158645 13	2.285195269443 83	2.255514254174 54	2.194398533108 3797	2.19575846517 247
PTTGC	1.873023706654 93	1.842425329965 41	1.869617487623 0698	1.882640989684 3002	1.848473709691 07	1.871300564484 54	1.870579774753 62	1.845543007837 798	1.86961859494 652
ROBINS	1.563266771027 3802	1.498842858422 11	1.567238141778 2402	1.515071719475 93	1.459287725116 63	1.506189607671 0701	1.565355246309 7902	1.500671877249 599	1.56713230714 488
SAWAD	1.511035251767 34	1.401536520478 91	1.449551245210 2	1.484211695081 8198	1.378937275569 47	1.429390708589 2799	1.520142014498 1799	1.408151266558 15	1.44768716888 394
SCB	1.507775140843 7199	1.462843995161 66	1.491649554876 63	1.467558586140 89	1.419472837957 86	1.463199043953 09	1.505565056156 88	1.460224255976 63	1.49200059418 767
SCC	1.137883785978 04	1.107318017756 18	1.129458719084 6	1.124338846404 1399	1.106696961048 39	1.129236545323 1	1.141667815327 79	1.104016086347 92	1.12952023119 1409
TASCO	2.490181520902 23	2.475554981376 55	2.425892854328 93	2.704776446404 63	2.715098703604 53	2.739168352544 35	2.488608167724 867	2.470366611520 73	2.42533998497 378
TCAP	1.187821008119 42	1.180252442728 37	1.180900934820 9	1.139412759403 99	1.133291572925 12	1.137247907069 73	1.187318124040 63	1.180771159758 3501	1.18088516158 916
TMB	1.439774463125 7	1.410465239316 1	1.440200986719 28	1.455230655378 29	1.441899257239 94	1.440765533895 1	1.441662344280 299	1.409740769386 09	1.44046212212 633
TOP	1.535932369463 92	1.512027514145 69	1.524713154339 84	1.538188479658 61	1.512745591116 55	1.554502110748 5502	1.534695217941 7002	1.512775049359 1399	1.52460409330 827
TPIPL	1.672877306218 68	1.670598566344 19	1.721220016345 74	1.940169244474 0202	1.821339221921 4701	1.734368064918 8498	1.677340013410 9302	1.672834798146 402	1.72211104447 598
TRUE	2.013888563396 28	1.993798880987 92	2.011096070345 16	2.038643933147 28	2.029070335821 32	2.019761966432 02	2.015233768532 12	1.992860623677 77	2.01062607322 131
TTW	0.822276514852 33	0.834453870246 1869	0.832319221296 088	0.859097885989 658	0.849024652810 159	0.862248951532 11	0.825461409546 672	0.831267813296 5421	0.83219075333889 7
TU	1.350912903872 4201	1.338147298645 32	1.382899845778 9299	1.394888696964 3601	1.352928473223 08	1.375898347483 2	1.350693285965 99	1.338046189652 28	1.38276313495 662
WHA	1.610570481963 08	1.548787476815 51	1.585410407950 6701	1.692302182270 2202	1.655799729889 5102	1.629769667360 09	1.629935948841 2097	1.544858824695 1	1.58574432534 3179 8

PredictPrice3 : EMA

Table 7 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	32.893910173672706	32.262846365349205	32.6926403066578	32.4000736719659	32.224442301969304	32.254362497507	32.92309987171 89	32.326557935877 04	32.97599447549 71
AOT	30.128136010893 7	30.781195586098598	23.124304178279 2	30.152124095693 3	30.576722800520102	23.130842112843	30.166763886822	30.762146519192502	22.515530004062
BA	2.759969383072 38	2.749580502382 3103	1.486369124771 1	2.702442483629529 8	2.678625869253 07	1.308736709186 59	2.735129242292 91	2.799839241748 703	1.56005198097 914
BANPU	49.749486276114204	46.861314675371204	50.769330007326 6	48.81134938699 76	45.448310711767 6	49.295374368645 5	49.298719855723 3	47.915178476735 7	50.96065785969 85
BBL	12.503370456901 9	10.488330628561 5	8.352423847503 25	12.497447909298801	10.887431872242601	8.278293700313991	11.988891914957 5	11.544322162534499	8.45904467250880 1
BCP	4.054273504049 94	3.778513038165 1504	2.139722622100 82	4.462051623171 8605	3.123896502977 04	2.203128262709 128	4.322931482248 87	2.394900871664 38	2.13569551547 237
BDMS	19.645208111671 2	19.754424964 953	10.580635510602 8	19.639597361303	19.703725934327 83	10.423474074399 799	19.593357526025 5	19.614927195952	10.14116661203 52
BEC	54.065642137114 8	53.156715938521 71	28.334869894637098	53.465660839874	52.094176256409995	29.253912562663 4	54.040288018656 6	55.715324112665 1	28.64649898893 08
BH	11.665500118037 5	12.835114173887 598	1.751681264425 39	11.823245288428 4	11.471298352877	2.083763276757 11	14.088988977232 299	10.446800776617 3	1.94076851079615 99
BLA	22.708040601524303	22.701922284418 9	9.275836977640 57	22.792083877803 9	22.805816754750598	10.420305071168599	22.72353443346296	22.163736297231303	9.24896050991 35
BTS	5.414747344070005	4.269520651610 5	1.520192794246 12	5.198776712155 27	4.788080882710 87	1.476671660634 87	5.460267816494	5.038019412172 65	1.54221424352 132
CBG	27.971787784230802	27.760201548605 4	27.085680787259 5	27.746707247291603	27.766707293431704	26.970957033679298	28.014575142371	27.857442155123	27.23520792031 97
CENTEL	16.526890549357702	15.436236498029 1	12.660261168676 8	16.976051230158 7	15.882634295516402	13.464286499369699	16.399446018396 3	14.464472524440 5	12.54540401604 2799
CK	4.531106372318329	6.512900122481461	2.418297412433709 8	4.685612997341 39	5.99597044343 19	2.537240243989 9098	4.998838548553 75	4.994809952564	2.39278515890 19
CPALL	4.192165548857 94	3.377125875868 82	2.220137411847 32	4.749338141569 3	3.710278071702 85	2.185330187482 52	4.773138155791	3.887246419473 05	2.21776123405 468
CPF	16.400345867522 6	15.946257396341 402	6.537459315152 249	16.779218763237 2	15.607408714091 6	6.151182610749 23	17.090488076709 402	16.264849855802	6.43812992244 02095
CPN	10.821407176784 199	10.743176236504	10.297730206266 701	10.686699701119 9	10.770231994677	10.465660382273 299	10.709240123968 799	10.654868834601 1	10.14423485613 62
DELTA	3.765950824 82	3.964672769950 82	1.857635035061 87	4.575659650833 5	4.956304934014 39	1.875610423786 298	4.027266110536 81	3.836197483015 07	1.86184328438 2199
DTAC	150.3011056125 93	146.5100491988 93	134.501745154603 01	150.731630670 352	148.415939119212	135.444595765041	150.334378673 281	151.354038161 87	136.941738125 046
EGCO	10.102733097972 1	10.095060807143 3	6.330810012587 05	10.063119713559 6	10.011003163755 6	6.268815505017 29	10.194891575294 7	10.051008272506 6	6.33195962572 631
GLOW	4.269985008827 35	3.736864824519 41	1.854621116663 44	4.449205578909661	4.186254786577 97	2.078533191296 49	4.406371201049 38	4.081211497748 43	1.84609595615 6149
GPSC	6.454119333689 77	7.441443016113 96	3.415484470058 5596	7.056430051595 49	4.411161810103 921	3.852872230590 25	7.182143790301 95	6.612431067251 65	3.62772316788 20097
HMPRO	8.453854890979 97	8.078267963742 99	4.746443434773 131	8.234929147868 92	8.300765075535 23	4.845411645147 7	9.039107974919 47	9.337718787730 02	4.73392621982 002
INTUCH	29.452914874410002	28.922984917302	29.055365242032	29.539644656717	28.966905022585 5	29.112300498257 8	29.462906763377 98	29.287846639288002	29.20796392127 5898
IRPC	14.400778101948 5	14.266317584386 5	11.642040967387 2	14.388080393091	15.013311983476 6	12.015093997953	14.797900874567 1	14.446299880872 4	12.00880255932 901
IVL	10.392229093042 7	9.469313835191 72	7.574301199781 56	9.934142968585 75	9.103533075671 71	7.954191331268 06	9.912699574213 491	9.396332106090 96	7.52041863448 252
KBANK	6.320835056030 09	4.160348075260 3	2.163596974553 72	6.871816048670 08	3.633615407195 89	2.270644604052 14	5.186691822000 769	4.638726954546 39	2.16875470222 50198
KCE	39.944314683220 25	41.011552489003 5	31.397200168762 2	40.429023318460 3	39.730642177375 4	31.304533786158 5	39.761258569999 605	40.454011567009 805	30.95404225512 714
KTB	13.338494686917098	10.400576213969199	4.005986936808 74	10.266856134173 6	9.878307906684 09	3.974280941954996	11.069823405007 7	11.481015535505 15	4.02819762598 877
LH	3.188850241995 59	2.669274441651 21	1.868396554518 6798	3.937191769680979 7	2.891716927576 8	2.135554662566 08	2.773741935369 4	2.589024578476 2	1.84030628989 479
MINT	21.734085900868 93	22.350535842193 8	13.625481609221 099	21.927980727638 9	22.353656469401102	14.622983889649499	21.586792287711702	22.349588393873 1	13.35962379652 7401
MTLS	4.283389840875 24	3.775882882016 7305	3.225737402370 3695	4.162493979314 39	4.112271341594 8	3.644754307318 0597	3.991187483461 2995	3.222090018463 08	3.23190595123 64
PS	3.069451426527 3	2.584438400191 94	1.340809314795 3	3.172736303349 8803	2.865540411637 47	1.342978112235 62	3.532709363270 24	3.536634722529697	1.34625415524 365
PTT	14.528076203899001	13.934836182802199	10.455439132048799	14.6858141378417	13.792719498650 2	11.006385810685	14.542070458370098	13.119002968275 9	10.64594725772 78
PTTEP	62.946941762089 8	65.375897516137 7	19.578917915981 6	63.096857220824496	65.332655243139 7	20.750946805700 1	64.918139497152	66.530129235504 7	20.95227352303 729
PTTGC	8.605306779715969	8.320450715376 17	4.230728085553 91	8.291696593183 34	8.553249170449 63	4.927327624886 69	8.337350411870 38	7.036143050114 3	4.16215652749 646
ROBINS	11.169257874109299	11.624436441236 2	6.070970427390 03	13.057209695423 7	11.218028683043 6	5.889676585259 631	12.433883247131 15	11.802734260768 6	6.09317270763 209
SAWAD	6.603463246392 93	6.182425805186 2	1.992065264711 9401	6.735406626205 901	5.954046678459 63	1.666526944038 4298	6.635596689709 93	6.088458992907 84	2.00863772883 699
SCB	21.097857369824 7	19.881804169601 9	17.282619220377	20.338771610446297	18.516442707265803	14.946129283490 6	20.884778272986 8	18.913354502349 1	17.25408744463 63
SCC	4.596971653394 21	4.124510731571950 5	1.262194330921 2	4.682525031331 42	4.056360191414 58	1.258839801624 11	4.706450422721 7	4.518953465488 05	1.26891639571682 01
TASCO	23.912255373263 37	37.910635777634 9	9.098108336644 34	20.127730300193 2	38.250683184766 88	11.859363179417 5	25.404691962177502	38.494408083731 15	15.62736990818 892
TCAP	5.676369822953 45	4.029232002483 9	3.093154586492 58	6.093215973788 25	4.313270202887 26	3.368101150380 3195	6.197882834510 03	4.803135985510 29	3.07433012349 232
TMB	2.687295287051 24	1.919935979594 16	1.637743263766 79	3.008298747340 6897	1.920602615780 12	1.651155503565 299	4.286473866116 65	2.336891655409 17	1.64271310861 45599
TOP	19.920312599604	19.026286540237 2	17.687998876017 1	19.795882357727198	18.162907330811 8	18.868292513407 5	19.684511255536	19.041803380722 4	17.58758684961 84
TPIPL	6.543430072125 67	7.175973047836 94	2.167119103567 6	6.676579080045 9	7.343783473433 36	2.001326562699 44	4.473370609669 92	7.124263454582 2	2.15184177847 484
TRUE	39.290988545806 7	40.077732622985 8	4.300161313155 354	32.712141893502	39.004389501771	4.070894037068 95	39.051219832143 5	39.265963151867	4.32110406957 895
TTW	5.797142256685 42	6.007068473830 45	0.902168964745232	5.936361668546 02	6.070542061452 3	0.989712349725 71	5.794732231200 1	5.180269877152 75	0.90122693710133 61
TU	6.794177484958 44	6.762209398890 36	1.699430151430 9602	5.148238155888 39	4.946473950123 3295	1.861478203904 71	5.514569254151 35	3.344022192146 38	1.69652972388 274
WHA	6.034148709080 17	4.054867114538 42	3.319996784616 9	5.093993088297 86	4.226525014067 36	2.637493421923 14	5.287487051359 98	4.223295875771 88	3.39123390266689 6

Table 8 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.7177896887058002	1.72727252745079	1.74930088236319	1.73013746330006998	1.74688927563200902	1.74456268144077	1.72058413243273	1.7251814059355102	1.74586971915374
AOT	1.04102738388466	1.04553109827124	1.09848978193752	1.05146384908242	1.04825394078324	1.173176698413	1.044713889034535	1.043871458525275	1.09768475342503
BA	0.963925607881193	1.00023155781622	1.17838812342031	0.998588168055203	0.908401973465159	0.966621376981 24	0.964334842479304	1.00546335508555	1.16463936040711
BANPU	2.05131958438605	2.00524237112314	2.14761953426973	2.132008921252 18	2.11530403783609	2.20104911256654	2.04991161661028	2.009062364216 32	2.12702790205739
BBL	1.2458561322555999	1.22912463769168	1.27487964729954	1.23504409148143	1.23220647222758	1.240568601080 45	1.243961048077 9	1.22780163765913	1.27065166326475
BCP	1.3945110972130401	1.36064251132229	1.41744488940652	1.423125163161 15	1.41829902639861	1.55197673502852	1.395611095405 61	1.360355362698807	1.41655346434848
BDMS	1.010596732929 28	0.969473811818044	0.894687943239 85	0.913753518225574	0.934705043760996	0.929756854971891	1.011011978509 18	0.998199312817491	0.893873717025034
BEC	1.68808423105143	1.73110816504602 98	1.80944359316541	1.693222661837 9402	1.67845138033929 02	1.669296720715 2001	1.685986166032 4702	1.714536125417 7802	1.83587253737 90699
BH	1.37634376164566	1.37621596977224	1.48752847256572	1.43037528314898	1.41724647172486	1.545436266806 6501	1.372791450629 04	1.379477865958 81	1.48971465542334
BLA	1.65955865685457	1.66426727621896	1.78604742118816	1.702457433705 5403	1.71479509263 749	1.734033005354 701	1.667354112033 98	1.673061541473 8599	1.78660539393 01603
BTS	0.973151964301428	0.971934226752 29229	0.95261707548327	1.00184316976816	1.00425503930995	1.00184177629204	0.970011456114 1591	0.971575370882 433	0.952174905942 8869
CBG	1.9776951764047	1.99190283550666	1.99105439536686 98	1.9839423698065 102	1.99383253805 44	2.04701509422655	1.971314947447	1.982846121686 2	1.99242814256758 98
CENTEL	1.581117602077 45	1.55301294050373	1.77665344365274 01	1.535724075354 15	1.55021414369425	1.555785400 44245	1.579515361969 23	1.551467814284 29	1.77131015723966 01
CK	1.432877635071 35	1.37360881157237	1.37333483448 16	1.382324599360 35	1.37879246741 4301	1.393080893863 246	1.395169203698	1.375995549877 07	1.37268545064 241
CPALL	1.341598255523 23	1.34054121155298	1.36492499194657	1.337208011454 93	1.34113659310 79001	1.332524666781 54	1.344289851815 08	1.340181486622 5	1.36383016744185
CPF	2.047281251422 66	2.0304838391621	2.08555971538417	2.01432820113 46	2.01628374027217	2.040128415092 02	2.052104384203 31	2.030654802750 57	2.08466742125798
CPN	1.2346973622008698	1.25767755469819	1.25308008173969	1.24441414852317 99	1.22713888212735	1.218487141 2927	1.238992258457 95	1.254807309204 79	1.25197940314624
DELTA	1.8262670935254601	1.82297769060863802	1.84687601134535 02	1.76609490955172 701	1.767887283757	1.758978061736 246	1.835208582862 201	1.82524908847 87	1.85342510116934 01
DTAC	3.2179570150427	3.19653493512455 98	3.31650587924078 04	3.12829393923968 204	3.12963705115 994	3.191294829339 91	3.210346580753 57	3.186755792506 897	3.31377377383308
EGCO	0.924533830332658	0.935744735250805	0.994370131977367	0.912035873377276	0.937768750700978 1	0.983612724230 4	0.924996839241 33	0.936248772429 5	0.984441825570 44
GLOW	1.3306972188322699	1.30580760388327	1.36952493074294	1.374155988806 92	1.34078508746792	1.429831399964 98	1.330179058178 54	1.312025866917 47	1.36966684000729
GPSC	1.4299495742309498	1.44656324633572	1.5500170401986	1.755876186810 54	1.77912238407653	1.913318248488 82	1.456395564590 9	1.464027894915 58	1.55507878111947
HMPRO	1.512149281809 44	1.51108579619544	1.51844007575761	1.484984886737 92	1.48935330753188 99	1.507640838060 45	1.508149764861 899	1.507227729128 87	1.51769347061087
INTUCH	1.682736763915 55	1.70627774079372	1.67429316045741	1.655533069944 44	1.70249962515211	1.721682256857 38	1.648258612333 4801	1.672202377107 2798	1.66692506681 25999
IRPC	1.369056005249 02	1.3108440141764	1.34407149947248	1.344128249698 61	1.33157904740749	1.364125233690 32	1.353413371451 15	1.309359816694 52	1.34778604035266
IVL	2.151329797474 78	2.13648243373973	2.09316063608786	2.087465495739697	2.0943306857473	2.091802154253 37	2.151873494678 35	2.135042403059 13	2.09096946900300 03
KBANK	1.450518878906 51	1.45766648519295	1.40561203795084	1.430382349236359 9	1.42488659851946	1.406617704920 36	1.463013686774	1.454534171852 86	1.40620562624141
KCE	1.471617702510 12	1.45969970152687	1.51325904631599 39	1.460396288935 24	1.45178197333801 098	1.456506674428 49	1.469559065557 3	1.459703754200 16	1.50988469573527
KTB	1.220650779518 69	1.19801086037182	1.19327519017535	1.205735040553 96	1.19165827074655	1.193300556032 27	1.210007471599 18	1.183807112124 67	1.19328995382839 99
LH	1.256696646968 33	1.25835375086968	1.25572652677189 98	1.2009317401582	1.20394674813612	1.211871252272 12	1.279673996573 602	1.257602920024 81	1.25691382930507 99
MINT	1.640161534736 57	1.66693634756311	1.80771488329857 99	1.653705523577559	1.65718763209764 01	1.766571063318 87	1.631894187081 09	1.674742264590 9602	1.80417087989 13
MTLS	0.957918667545 31	0.948681229589 079	0.948890024805 1231	1.041234279434 69	1.104486693452 0499	1.065920009593 898	0.949446969407 405	0.944555062557 9009	0.95060252252 28221
PS	1.31587285847958	1.34980330483361	1.28929485064658	1.318060126778 3598	1.31703071945001	1.353714265849 48	1.338951059345 9502	1.354021671952 702	1.28788053299 793
PTT	1.8812772539120899	1.84716904775098 01	1.83032093328196 99	1.856746363684 7101	1.83124042236692 99	1.846045476583 98	1.865334225443	1.844738612500 102	1.82326595802622 98
PTTEP	2.3159593158663903	2.26644420228858	2.24241809338227	2.226085376483 51	2.21608561942569	2.241317713529 6	2.296672906506 94	2.260298022907 2966	2.24061731124146
PTTGC	1.825640950973 68	1.84227306730111	1.80292880391541	1.911081554686 61	1.86434944438781 01	1.787241798765670 1	1.823512838651 82	1.834323051508 44	1.80241129889273 01
ROBINS	1.514156649423 46	1.51038225523094	1.66463946892222	1.44756094625 96	1.45089576573 64	1.506036752841 1	1.503416223378 31	1.508261667399 6	1.65683469617404 98
SAWAD	1.572523149145 66	1.55011920266421 99	1.50120965311863 99	1.421579684424 97	1.43874978124 999	1.482762644980 31	1.603095018747 74	1.511489539776 2	1.502080730261
SCB	1.450986325112 08	1.42352956674879 9	1.47547816365210 02	1.479378229782 69	1.46523818933 696	1.465935667475 1	1.445243751990 04	1.423479757829 85	1.47811126506860 01
SCC	1.159528410359 64	1.09623951776	1.09028004933678 01	1.107523469559 39	1.10392214204306	1.086927596507 998	1.161088481264 98	1.098414039398 57	1.08934951902626
TASCO	2.48723536335 58	2.42585670416386	2.96069083108713	2.503441123757 7903	2.46302690520766	2.767201018159 75	2.471634127213 4103	2.420666056138 41	2.92914151100671
TCAP	1.1899703358666 3	1.16826964557632 02	1.16586254694539	1.171385589982 2	1.14245025195935	1.145652746087 65	1.192751423781 2	1.168419304938 33	1.16626152809041
TMB	1.5502315385016598	1.48534274474467	1.43023939105754	1.519346591494 47	1.48862244213054 98	1.494464051506 3299	1.550179006129 99	1.508707717065 7799	1.42981191518495
TOP	1.5411627340333 3	1.53797483201795	1.57024708069381	1.526057760298 17	1.53235169881639	1.536666663941 92	1.525632283204 73	1.529953380390 7301	1.56740017714003
TPIPL	2.005629749813 32	1.82467830285583	1.76954004714128	2.027173150270 11	1.87442797634637	1.676057218029 16	1.980694664176 3	1.901856922089 74	1.76757029935674
TRUE	2.112911257668 36	2.05856842527166 03	2.00987218682427	2.072819892682 17	2.10398107067717	2.064311663394 15	2.102925485231 4097	2.058029287241	2.01037997088802
TTW	0.878636522466055	0.80831464563908	0.81161740489457 59	0.827324876752 0849	0.842461024913 8169	0.870821678352 899	0.875444717983 896	0.808566738398 497	0.81140558084769 7
TU	1.326777195237 499	1.31225435450075	1.37384301541704	1.378140837760 94	1.35649742221292	1.326453915034 9899	1.329587509702 17	1.314579008821 74	1.37280225804 37
WHA	1.731487295454 05	1.65140986990350 98	1.54353629398764	1.920283464222 0001	1.83564075980677	1.707046137355 75	1.679144673378 46	1.692838590533 73	1.54482546254 85

Table 9 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.7232101809702498	1.731499972608 02	1.73173037078859 98	1.72764387969 08	1.74079307929739	1.75154249545357 99	1.723431675539 84	1.730158342648 7799	1.73167372976768 99
AOT	1.056753927812 87	1.05807955335244	1.10713799547282	1.09957170893932	1.07211933334955	1.103239958871 24	1.054815307272 04	1.059952909854 26	1.10840168302 87599
BA	1.012978975573 32	0.983966827270 3	1.05832532990766	0.963265778504828	0.920200620404723	0.927444745490199	1.044412028095 701	0.997218446097 875	1.04625705894804
BANPU	2.068008143488 08	2.024903518222 06	2.13030680049721	2.142822610852 78	2.11054762231694	2.213171352714 93	2.065470646372 16	2.022471461907 45	2.11234154512973
BBL	1.245530650935 7	1.22760558995263	1.26826377673002	1.23680926440633	1.23005212911232	1.231579876502 74	1.244884267539 27	1.227411500798 4598	1.26608337353365
BCP	1.391447656530970 1	1.35936833384982 01	1.41744488940652	1.462081600984 68	1.41881810377202	1.551976735028 52	1.394450769235 96	1.360260222945 602	1.41663661478677
BDMS	1.00357207265081	1.00233557133895	0.89482545975945 99	0.919563215519807	0.916424586278829 1	0.929764984759329	1.010266826695 93	0.983398523681 844	0.89400214990252
BEC	1.669300131186 88	1.66062442948941	1.7516812113421	1.667838281960 86	1.67808247594826 99	1.684899538688 69	1.667387932658 5601	1.668901775911 726	1.7620296436162
BH	1.386712284148 89	1.38502343371567	1.46687861360528	1.380465044622802	1.37780755112318	1.425293062943 67	1.418577245502 2199	1.381199663418 89	1.47389573627174
BLA	1.664845076747 19	1.67850688026809	1.78615069662587	1.699749397079 93	1.70917774790705	1.733983071824 87	1.660886770894 9502	1.677139924421 998	1.78645069489482
BTS	0.968641051656636	0.969610946535083 1	0.942207185104827	1.011481390772 2902	0.987568135009849 1	1.012889297817 84	0.968728231649 642	0.964516631490 848	0.94307229966254 791
CBG	1.968870407386 3899	2.00812800858825	1.99105439536686 98	1.98250905342 949	1.99453459945 6498	2.047015094226 55	1.970761172648 53	1.985725336085 698	1.99242814256758 98
CENTEL	1.595312383878 84	1.54831569365184	1.74950778410335 02	1.554824630978 29	1.53416946106727	1.520653478190 07	1.590095469719 1	1.550730308449 87	1.74278511671672 99
CK	1.560764322645 3498	1.490389732220 9	1.38987504119188	1.423209301324 69	1.38743600160865	1.449415283197 18	1.499650002270 07	1.456662670678 68	1.38795423001349
CPALL	1.344135467641 7299	1.33950077364262	1.36492499194657	1.338532950854 81	1.34106346709043	1.332524666781 54	1.341542738128 71	1.340078001631 07	1.36382731528070 99
CPF	2.027508501629 14	2.03628162097267	2.16057258368859	2.024992230690 01	2.03788106351729	2.143641759041 66	2.021831402292 75	2.026997496518 53	2.1655610008806
CPN	1.230470930885 35	1.25774821518383	1.25308008173969	1.244219603283 79	1.22640684228235	1.218487141 2927	1.240436373095 299	1.254111407378 78	1.25197940314624
DELTA	1.834894589102 6698	1.82369300818424 99	1.84687601134535 02	1.759012929022 99	1.76084438164662 02	1.758978061732 46	1.831520809980 38	1.826518654049 86	1.85342510116934 99
DTAC	3.159698823904 97	3.13124894073303 97	3.21943153988237	3.109754552082 6	3.11164410255 14	3.148886390204 92	3.160111190945 4	3.129778309156	3.21257618545 48
EGCO	0.925477295066793	0.936714990838096 1	0.992658720160105 9	0.912467685081 386	0.925918195073755	0.983192656052418	0.924701683909 13	0.934761507935807	0.98291605737787 5
GLOW	1.328509958966 589	1.31183248302292	1.38138521999811	1.370615462161 1901	1.34058643744072	1.429820616875 35	1.328939987372 32	1.313619189231 121	1.38140215397409
GPSC	1.458417383473 1	1.44793056309917	1.5500170401986	1.759968432750649 7	1.82392755659781 01	1.913318248488 82	1.450082473754 67	1.488455842464 58	1.55507878111947
HMPRO	1.513834645640 86	1.51180503156 33	1.52206419870868	1.485931683810 15	1.48984980809 98	1.511071591033 82	1.509456040054 699	1.509172558730 4801	1.52129335739763
INTUCH	1.655576329499 3	1.70244870856429	1.67429316045741	1.664181320553 98	1.71430364921081 01	1.721682256857 38	1.670034728517 6502	1.673003092617 5699	1.66692506681 25999
IRPC	1.377337209383 33	1.32742987871242	1.38072171877 58	1.383597930844 98	1.36156462685551	1.413868612035 09	1.377118956364 72	1.328705112149 77	1.38571327550963
IVL	2.130015220005 26	2.16544394806715	2.06966476372226	2.089213870723 65	2.0940448717825	2.085795913665 39	2.126407440772 88	2.153640039730 27	2.07112464057606 97
KBANK	1.411385887176 1	1.43404579623132	1.40872372041379	1.415470067132 95	1.40948678915784	1.421239665516 2	1.413930945468 85	1.427441640787	1.40807275961519
KCE	1.472761673611 47	1.46463437553257 01	1.50994426335782	1.465980351920 43	1.45357524302936	1.468562797492 98	1.478149103780 2998	1.462721150375 83	1.50805838884955
KTB	1.223911096177 8101	1.18592242524031	1.19327519017359	1.212096163982 63	1.18801654672153	1.193254243767 27	1.219007997493 75	1.189590038767 77	1.19326312146879
LH	1.243366392647 51	1.25411781150813	1.23313725290826	1.217382911130 46	1.20468378225939	1.210569612254 1298	1.247888903562 4199	1.255353330307 98	1.23560126000666
MINT	1.659894298183 32	1.65240736525998	1.81276984794876	1.676959141619949 9	1.66173831835852	1.711435150765 75	1.633834188466 602	1.649960311115 72	1.81285740449439 98
MTLS	0.950950712763002	1.022933281707190 1	0.948916007272545 9	1.024262568697 15	1.08640738129 19	1.065920009593 898	0.948225752695361	1.011075169112 7902	0.95057743896309 3
PS	1.331689931067 89	1.33075172093806	1.25455394317622 98	1.360994618231 3502	1.35362673355 5	1.271401442526	1.341185751532 9901	1.308540123813 101	1.25531185629616
PTT	1.879037553257 89	1.84129986786952	1.83206824843561 99	1.864540812607 85	1.82886287705 1	1.846512979560 13	1.873396450744 58	1.842500471667 198	1.82515065141583
PTTEP	2.239337664718 498	2.23744850047048	2.2598746290282	2.216603362326 99	2.21133215412 433	2.224740372312 7	2.266354361225 32	2.234313346955 82	2.25416877005029
PTTGC	1.837960068781 11	1.84160326186 95	1.80292880391541	1.911168816875 25	1.86474545302333 98	1.787241798765600 1	1.814181844609 4902	1.835875926846 34	1.80241129889273 01
ROBINS	1.508754886297 6201	1.50979926771 36	1.66463946892222	1.441965453026 76	1.45827538964417	1.506036752840 11	1.510638505809 02	1.512045765525	1.65683469617404 98
SAWAD	1.538969351266879 8	1.54843493006098	1.50113673503325 98	1.444125735395 4	1.45772649155267	1.482762644980 31	1.539290062091 19	1.497789696895 19	1.50212252125452
SCB	1.447183986510 699	1.42780091901743	1.47547816365210 02	1.479886408369 2	1.46568939731751	1.465935667475 51	1.444124683120 8	1.423120060617 01	1.47811126506860 01
SCC	1.168589017332 21	1.09964075597049	1.08925285734519	1.109603428417299	1.10965291019258	1.102025739736 87	1.163675850621 08	1.094385962460 97	1.08830336926800 2
TASCO	2.726214888962 357	2.71647624921523	3.51392898386209	2.781247857036 98	2.73803016808527	2.886441122279 07	2.714401989611 38	2.651885909506 4	3.46276438937503 04
TCAP	1.193807708608 2	1.16835007689257	1.16586254694539	1.168341856863 58	1.14374866915 168	1.145652746087 65	1.192829326881 85	1.166914513741 75	1.16626152809041
TMB	1.543337837337 48	1.49368392514265	1.42988568851858	1.526667874038 95	1.49476796867963	1.494757290814 4	1.551380479946 37	1.477779533307 92	1.42923968640366
TOP	1.530876908282 12	1.53527869772005	1.57024708069381	1.517780861597 07	1.53065203254796	1.536666663941 92	1.522930873632 0102	1.531037591069 7	1.56740017714003 99
TPIPL	2.252915780990 7	1.96510169239 19	1.74512774703446 02	2.008068678283 1	1.97981820064146	1.897094928940 08	2.025626722164 77	1.986504511387 9	1.74560117510175
TRUE	2.036706154686 81	2.02111451194 59	1.99344004033737	2.016193767343 28	2.02168393424651	1.989626612920 0001	2.033399110655 77	2.015725893597 198	1.99303139992 646
TTW	0.873963914564308	0.808837844300295	0.81351055337214 01	0.825526399762935	0.832979336970128	0.876629874491343	0.871210153458 8269	0.811582337493 8611	0.81350312362 5307
TU	1.318741978641 97	1.31927509457616	1.37384301541704 02	1.371650368402 99	1.36072084904261	1.326455054204 0499	1.318800086106 1	1.309230662361 56	1.37253770151654
WHA	1.670855324912 46	1.65332274766991 02	1.53236144365465	1.874490980724 23	1.85199809482212 97	1.708157440346 44	1.704601788407 001	1.592466427870 96	1.53306403153553

PredictPrice4 : Volume

Table 10 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	32.149164266709604	31.162658712159104	28.7251125111676	31.7945406831822	32.4833616054197	29.267701917391797	32.020726303811195	31.121202918231898	28.9273118474747
AOT	28.5267798692923	25.74215487275	15.0807784671472	28.440453325076497	26.41361283261 74	15.7429075940444	29.02679388832249	23.711355394160798	15.03938926662 82
BA	0.915642079067347	0.9764919785688491	1.056478080151005	0.9496960793945359	0.9233885709080004	1.001321421841 35	0.959748416795 95	0.976337729819102	1.055209055741966
BANPU	40.5697202474838	31.21578771317 13	15.734999004751298	41.0869711722734	35.910540272200194	20.36311755401 48	36.33208654668 71	32.019991819207696	15.824131747303898
BBL	6.1841542739718705	5.399522713640 43	3.2280600577592704	7.05890243336417	6.004655367497691	3.6981186146304696	6.053122938 87988	5.25934271674415	3.2196366682712503
BCP	1.3487249815026	1.5030266160186099	1.4184815496579402	1.5352458433471499	1.51546819350446	1.5092807699957	1.351468731834399 9	1.444229997763 52	1.4174332371454201
BDMS	14.8408945270543	12.8258553147978	6.187577409981619	15.1059547027521	12.592007539851	6.5033555646109 79	13.050762216182099	12.21677707640 01	6.121952875046
BEC	32.8486641704839	40.877676434966	15.602397354328602	34.9910356114191	33.778216584056	16.417030311661097	34.3493116403111	31.475555590233	15.72116837244 99
BH	1.4903935615142498	1.737992293075 06	2.1536226976073	1.482834948178 04	1.7149595328065	2.186414975474 5	1.490378436729200 1	1.807678509485 43	2.153654510689 05
BLA	17.099999203587597	17.103410787951	7.675690662378 44	16.983633005 2395	15.545063619985	8.949897479266 39	16.77655243606 583	16.110454896579	7.602945141495191
BTS	1.0303330604396401	1.143785966900283	1.048200073095 75	1.12925001512555	1.23289997069 94	1.1122834631276102	1.034790094047 36	1.129649891517 22	1.047701961181 52
CBG	28.30103366059848	28.097671335612002	27.31888000963989	28.349960870211397	27.6084691488077	26.2442167188702	28.158152752861596	27.984301821491297	27.135681986530702
CENTEL	6.2560000462776095	6.2658698996006095	3.4758249056365798	6.262674247890169	6.564042305562009	3.9860612355212 28	6.660624722233679	3.847871101363 41	3.471116686877203
CK	3.916374692071 51	3.634883489250 64	2.328609961651 84	3.404824453464 94	3.901233528564803	2.405190587959 77	3.777429274753404	3.947535050494 9	2.306088952308 23
CPALL	2.342284339389384	2.238860633297 23	1.738721639391 902	2.414752076384 17	2.50938212205377	1.961449137539119 9	2.170203682466 1897	2.246289631431 99	1.740559390897 6997
CPF	3.973479990609396	3.942136222225805	3.301926799530 4302	3.709703554496 5102	4.327541895652629 5	3.44070455654 43	3.922953648887 874	4.271902873762 23	3.324027373555 85
CPN	10.69778535489 57	10.358198276238 4	7.522488848018 757	10.9029232487305	10.421367599068999	8.770623137 257	10.501375388567	10.540146260594 6	7.471519479781 399
DELTA	1.8694657024272299	2.057163039047 48	1.923309232355 17	1.89361590809848	2.067235179858 12	1.9008739975586002	1.922424837321 77	2.069865246435 11	1.922568657939 5
DTAC	131.90355913134098	128.59621058178197	57.292153159357795	126.258213144266	113.505382513662	63.26170353251 05	134.132364650516	105.669674128328 99	57.911805388245696
EGCO	7.699816489899 64	7.891547052481 15	4.187003496496 73	8.152465058502969	7.816009325582971	4.367100367471 95	7.904794174566 34	7.347305230075 89	4.158124516953 14
GLOW	1.319410510268 25	1.445472122768 138	1.302132702340 15	1.483141397923 65	1.539122828352 82	1.410491960863 36	1.352330792052 98	1.385339292131 95	1.301130027517 3598
GPSC	2.949921611355 53	3.991696777674 5795	1.940644990400 785	3.517950987494 7397	3.492197922965 57	2.69512395444 79	3.737495459726 76	3.247830148280 08	1.971855729505 73
HMPRO	4.865385374363719	4.983038083862 399	3.532468471861 42	4.953396162388 13	5.177569781641 96	3.559635670530 3296	4.844112290747 82	5.115039744530 86	3.529778104148 35
INTUCH	28.7861011955993	28.0006075500558	24.818219021804 4	28.802150921267998	27.144086249715098	26.607319045225402	29.021269918217502	27.843250528883 8	24.932129771737 4
IRPC	9.163545299594 37	10.054904194134 5	5.245851896265 89	8.826882591152 33	8.684431009772 22	5.145988650571 13	10.666926705243 8	9.011284988603 329	5.253076637173 21
IVL	8.336129435594 03	7.844554309852 41	5.616195357117419 5	8.621526047264 97	8.52894837177522	6.042512310276 48	8.457772294486 65	7.251544929795 729	5.605677411073 12
KBANK	3.168610113072520 4	3.238151378190 45	2.358365541951 07	3.347913140353 72	2.999706306125904	2.382573907594 88	2.742604782609 9103	2.984993866944 371	2.356325042394 8903
KCE	36.2500927965589	34.717249832685 4	23.7834485374 39	36.7422058682992	35.763943593443095	24.527058953755	34.892362151917 7	36.081711323412 804	23.671671414400 37
KTB	1.4660876450586602	1.8518022457141299	1.3034461923973701	1.6997360567537 7	1.808098641888 73	1.4603653062125501	1.476057716765 4199	1.739571681645 33	1.305714491482 52
LH	1.278666460982 31	1.433022944906 44	1.345720037798 62	1.338166229218461	1.427042070080999 8	1.364177940670 17	1.261712568026 43	1.457026216128 2	1.345878730014 86
MINT	18.484874791038	15.152654822055	9.129781245675469	18.240298119613 5	18.31147253584 65	10.018447134509 6	19.373893168877903	17.322860066574 9	9.024157601354 29
MTLS	3.578942270340 71	3.150729472151 21	2.435444419986 48	3.303070480111 82	3.455789553380 58	2.721475071952849 8	3.424963427049 12	3.234698498164 47	2.434542026637 43
PS	1.360078844864337	1.607959021690 97	1.365990654589139	1.334076726501399	1.583747278763340 2	1.427137192655 4	1.367303327706 38	1.598199584807 03	1.364050620628 09
PTT	5.840878223423680 5	6.040776446951900 5	4.010208697266 19	6.945943229589 76	6.199420088630 01	4.618306384822 71	6.387947859193 93	6.476939173170 081	4.003612312141 87
PTTEP	21.41627639926 73	23.917555050750 5	8.141244228630 41	19.277540308099	21.212618514543	8.145441879568 27	19.550731987077 2	27.340691808501 4	8.150029699346 97
PTTGC	3.869243985940 69	3.833360576116 975	2.698470316026 76	3.898419763946 23	4.103853130032499	2.944715524411 057	3.937754024888 3	4.028224945304 13	2.689759738627 63
ROBINS	5.445883193676 65	4.956777705771 5	3.051610509037079 8	5.960574593749 98	4.871192112152 9	3.575006283419 63	5.484917132309 279	5.679145124760 33	3.045467772673 17
SAWAD	1.677291809096 78	1.622221261902	1.497189589214 798	1.686155357218 73	1.624238451799 4701	1.523252339654 06	1.577280707268 6599	1.635851032015 62	1.497048375213 22
SCB	7.826431346787 74	7.779493743903 459	4.842758863061 16	8.732044159480 7	7.595149658025 9095	5.356551577515 4	7.972903669853 15	8.620899762456 83	4.851544412497 96
SCC	1.119181815073 25	1.165985983062 4701	1.125410392139 44	1.133616319287 9	1.192438243107 71	1.126134999645 2201	1.117913535340 68	1.162689662647 83	1.125292288185 69
TASCO	7.474450274848 7305	7.731301647916 56	4.301494211658221	6.130247753161 59	8.204594637836 61	4.309577539846	7.021794214108 86	8.223476746380 6	4.328866584517 67
TCAP	2.645992395383290 2	2.551470430035 37	1.968207440586779 8	2.626128446080399 7	2.360053599081103	2.709278576367989 6	2.686212023796 48	1.966684605467449 8	1.394130482520 05
TMB	1.560863765912 88	1.742565652230710 2	1.462632964953 46	1.649588053413 66	1.655612755159 25	1.502780365962 02	1.568696568906 599	1.691822684296 88	1.462603727247 59
TOP	18.075152978152	18.413062416488 398	13.595888910866	18.539432035823 1	17.517312853185 2	13.949347135827	18.213712028346	16.501381310210 14	13.374699207061
TPIPL	2.081638064786 96	2.408853705266 61	2.177046491270880 2	2.106691332590800 3	2.457595961775 25	1.988935247469 7302	2.114043827378 67	2.383633026070 45	2.178361682287 89
TRUE	2.798347655767 83	3.286083410012 44	2.726343084351	2.821308890602 59	3.067211667206 49	2.837983595590 84	2.980491168824 44	3.340039137231 102	2.731781639241 38
TTW	1.020669703259398	1.027061071795 42	0.988434048737259	0.971909976516734 9	1.009687315448280 2	1.008457251868 66	0.927716223083 2701	1.031236787585 29	0.986516633427949
TU	1.372395811862 69	1.455078146888 96	1.394307407425 12	1.423964346438 94	1.415619938968 39	1.437038480122 0701	1.393261302574 6399	1.478887022421 9401	1.394130482520 05
WHA	1.662842630788 85	1.853146985895 96	1.732426723322 702	1.785449674507 67	1.859683940919 1002	1.836227788094 57	1.669471848074 4002	1.843764781076 02	1.732642519870 46

Table 11 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.769370598629 77	1.842039216014550 1	1.726987953814 12	1.801643202652 73	1.796945699925 8	1.755748339517 42	1.767737395900 01	1.783564529440 25	1.727162555669 43
AOT	1.075648168162 84	1.106599122043 33	1.043759647816 86	1.059916726412 71	1.060633822202 81	1.047194044291 73	1.074329420717 76	1.082624015011 07	1.043575354159 25
BA	0.956307931578244	0.912448301223678	0.938547975348825	1.054370394390130 1	0.994212636424492	0.936626210548136	0.947433565134189 1	0.909761571453565	0.939511272648872 9
BANPU	1.981505895651 57	2.011495493865	2.001550919826 01	1.975546936797199	1.990490212794 48	1.993255771553342	1.993978238394 49	2.026476730166 69	2.001565824600 93
BBL	1.245862396538 81	1.262269229896 69	1.232453442246 75	1.244855266395 67	1.233991334067 3798	1.224371646049 6201	1.262846582119 4699	1.273273264970 03	1.231829571479 08
BCP	1.396399944739 33	1.375783529767 14	1.347484286602 73	1.384052598763 76	1.438126940984 1702	1.340804909347 13	1.394497452123 05	1.376297844048 5498	1.347737194971 3801
BDMS	0.928483928366983	0.956094239716651	0.884679419070219	0.974044388238517	0.984724500335 7581	0.914886646237 8389	0.960883169057929	0.955377784532 2751	0.883687175008 966
BEC	1.732360501938 03	1.823611248360 39	1.680144011810 9598	1.816982106575 82	1.840987262470 1298	1.692220796830 37	1.733544504819 1698	1.784657584340 8101	1.680020053711 271
BH	1.423898081313 33	1.425852836260 56	1.410457933425 9701	1.522923142786 76	1.529848055319	1.469781914542 4	1.424140956450 02	1.425698049436 5	1.410658543209 9699
BLA	1.689690591516 61	1.708071393562 1599	1.649134463086 9198	1.700589729976 39	1.713181887378 16	1.723884601635 06	1.689538484649 999	1.709011121150 34	1.648726903638 24
BTS	0.969555591799869	0.959466157263181	0.918960932388268 9	0.974411474767060 9	0.965523482351871	0.968422715766 319	0.969100835371746 1	0.957106766962 304	0.918829221489097
CBG	2.035196485124 92	1.983833689237 93	1.994033911486 098	2.059175505554 3	2.034473624155 9	2.065896283798 65	1.991362178162 9702	2.027205387941 14	1.993774001811 71
CENTEL	1.498035892444 85	1.524021730406 65	1.480282666523 36298	1.562484004730 27	1.545302856775 33	1.541230196828 83	1.495359005776 82	1.527857320129 59	1.480108700393 46
CK	1.352509345608 7	1.382475593514 67	1.340929883208 64	1.334095017117 28	1.395692506624 41	1.347006720291 6098	1.344680923545 97	1.387019384452 23	1.340918141194 25
CPALL	1.313501022353 3	1.358714454945 46	1.326153514263 55	1.349330814458 5	1.349745942747 19	1.350570061139 58	1.335898275378 9301	1.341637188282 1	1.326225008379 3998
CPF	2.026749853292 28	2.096542426109	2.015693338107 4102	2.090279787203 007	2.079019262885 53	2.009016309849 39	2.026856239129 61	2.098984562020 84	2.015363494959 65
CPN	1.250125450090 64	1.265974906548 05	1.228271574890 5	1.240884683361 71	1.251430129817 7302	1.207321568726 501	1.274774731475 17	1.253261061795 01	1.228313090446 98
DELTA	1.732712870667 8199	1.740290945860 4399	1.712344616835 15	1.795597266147 95	1.760420345073 5398	1.734982884002 4202	1.732053532333 799	1.741184040287 81	1.712191521518 3199
DTAC	3.088192745981	3.208869135264 3	3.080173079776 42	3.336555205485 34	3.232975912234 7	3.146360606311 5897	3.130254779088 73	3.153793956200 35	3.078756572440 66
EGCO	0.956651453654872	0.965823122240155	0.912955910009281	0.968366373455500 9	0.956063212004881 1	0.920138400750523 1	0.958462449560 15	0.965274408227 6791	0.913338078435 806
GLOW	1.291716106705 96	1.304459627100 99	1.254789354390900 1	1.319494131707 16	1.310441845761720 2	1.263287753060 81	1.280937630834 19	1.344840081118 9	1.255078912607 68
GPSC	1.510081999947 79	1.538365205277 62	1.484495705069502	1.711113325873 07	1.775343592811 6499	1.755103226127 6598	1.512317168556 8502	1.529366840250 63	1.487046409467 69
HMPRO	1.522038625633 6	1.507753204448 38	1.464383541337 2501	1.498518847416 76	1.496148551626 86	1.454852383646 61	1.523151216216 98	1.508389662197 58	1.463364210807 9
INTUCH	1.640542266817 57	1.645850178721 47	1.603824605559 72	1.672611799684 2	1.658707422553589 8	1.616341010381 7302	1.644921676879 24	1.655986465591	1.603828210723 1598
IRPC	1.316204247861899 8	1.344356710190 95	1.280904096291 95	1.305115814044 79	1.346408730128 18	1.270485076213 599	1.316658006412 57	1.364132565729 1598	1.280092583082 28
IVL	2.194359094667 25	2.194598188901 05	2.100524264117 36	2.172859626324 498	2.184505360059 54	2.115676528762 66	2.192565056326 3	2.218359738924 9203	2.100870721424 13
KBANK	1.418246280044 84	1.421822278644 5399	1.378783388208 88	1.444634321130 96	1.448163997814 0302	1.417461038812 3199	1.415548064807 43	1.424614748269 79	1.378972291642 6
KCE	1.506452236847 3898	1.522361804214 08	1.456727275716 53	1.471477456298 8	1.480963113749 3	1.456478946365 15	1.507039421402 94	1.533779632055 39	1.456726211031 9
KTB	1.179655320036 35	1.217020233246 26	1.154822931274 8202	1.172259235661 81	1.159312561222 0498	1.179354533517 39	1.177736733066 27	1.220098725179 62	1.154636778017 3201
LH	1.248481113474 84	1.225027520807 34	1.213089108684 17	1.265067609546 35	1.266098676073 66	1.232985833045 09	1.257892559413 93	1.230470621936 47	1.213476390090 09
MINT	1.591564658934699	1.602697235942 64	1.571102867004790 2	1.588457528712 79	1.599294676326 3398	1.578316686585 199	1.581998037104 598	1.601005966392 202	1.570864410049 2102
MTLS	1.023537546979 98	1.026445080679 03	0.935414388222 879	0.978281343943 403	0.991002723754 638	0.999189779615 8359	1.041404108797 24	1.030083365376 6601	0.933724230190 396
PS	1.322228913012 73	1.392581973593 2998	1.242769710690 54	1.344243004839 74	1.365024278311 46	1.258495226976 99	1.337020425620 61	1.413686585849 57	1.242716191511 83
PTT	1.876194512702 6498	1.898341844332 62	1.830202672726 95	1.865542736194 45	1.884320445606 28	1.818431380970 97	1.852514448539 64	1.896471604524 63	1.830874916265 79
PTTEP	2.457630607254 03	2.507051981332 78	2.301654156690 06	2.214865841023 8	2.258953203463 14	2.166733449057 24	2.460909729882 34	2.514392522459 76	2.301911673338 4
PTTGC	1.881453498938 3802	1.858741777177 05	1.774512513024 2	1.852420285834 7099	1.852556581096 71	1.861878559707 37	1.862432981915 0398	1.864590863073 19	1.774539645612 76
ROBINS	1.532692484573 13	1.562643913147 19	1.461669057931 71	1.530993422173 07	1.538486516288 8	1.460909972422 23	1.533291716935 1599	1.589075135122 6502	1.461671251881 03
SAWAD	1.512347722941 31	1.526070171561 69	1.392697830387 47	1.533236214650 59	1.518576906596 02	1.410498264029 39	1.494004081726 101	1.514303131479 31	1.392246760617 4
SCB	1.468184829396 7399	1.493497135265 97	1.475027788004 32	1.455556602093 6102	1.461234572463 95	1.450802898893 32	1.460035929512 32	1.506911374699 72	1.475017079234 07
SCC	1.097696480516 75	1.151702358536 3	1.079824261094 2599	1.094948259375 75	1.114034880180 6	1.074752823604 89	1.095998074898 4899	1.119283400445 42	1.080134543253 51
TASCO	2.408767543388 35	2.385921616856 58	2.343327515203 36	2.431911803629 34	2.422608093841 07	2.472138279683 68	2.391172555586 544	2.383783599222 98	2.342904040011 67
TCAP	1.198367176143 82	1.174895546083 13	1.130871988907 34	1.144009499430 402	1.166905671670 8002	1.127682011868 52	1.197457812934 1600	1.176653331299 98	1.130587854286 3799
TMB	1.481873479872 19	1.460957428970 55	1.387185102160 46	1.453184994352 1902	1.483340335259 9	1.476042713095 57	1.462579434194 2702	1.495353746215 48	1.387072806389 82
TOP	1.516086700801 52	1.537597906119 49	1.493140036257 42	1.570617459881 7	1.574593939277 44	1.510710992818 79	1.525975576410 64	1.533356632238	1.493607441788 71
TPIPL	1.682133731670 7399	1.776562020962 568	1.591664429555 266	1.933206874345 26	1.999019243352	1.728638462959 6598	1.662849231533 0499	1.784988159421 81	1.590446835353 5
TRUE	2.079840240294 61	2.144592429028 76	1.973196333935 71	2.199165102220 2503	2.185542930949 22	2.068902760392 89	2.081786062270 7602	2.148331908871 5	1.979332091222 949
TTW	0.817999067217817	0.870593511953817 1	0.811569712425728	0.868844167473513 9	0.849259512240740 9	0.854050697853389	0.816138096637646 9	0.872261645081263 9	0.811352159732190 1
TU	1.353642867590 64	1.377416737115 03	1.328432664413 07	1.388644228442 33	1.376682907286259 9	1.333594806464 51	1.353618353281 88	1.373173290006 15	1.328103027487 4501
WHA	1.61043088852 81	1.655521813666 162	1.514624282993 89	1.647462123097 598	1.645525660715 76	1.570469150279 23	1.620749117103 6699	1.652682543844 32	1.514471064604 71

Table 12 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.7583277861165598	1.75588810780364	1.7461509379874203	1.7759959711569602	1.77077184150388	1.7550024382957399	1.7576044746449497	1.75531490865941	1.7445550291914198
AOT	1.07373419825567	1.09927116401715	1.0378001375988999	1.06188010519972	1.06564094295319	1.02253013333137	1.07536034881681	1.10685339376282	1.03792074342421
BA	0.9728375407811339	0.962979477921521	0.924350192720492	1.00912354899229	0.9874429518265591	0.9153887137494441	0.9684569759482959	0.961311968935944	0.9222192216938621
BANPU	1.99494741773841	2.00988930245872	2.00042759722319	1.98689562872688	1.95926402401304	1.99583559735955	1.9951124879858	2.01033646322751	2.0008590617705604
BBL	1.2576684450017699	1.26516728966045	1.23049319062815	1.2412455681011099	1.23789693302042	1.2238719530895101	1.24768981579503	1.26903737387328	1.23000572139383
BCP	1.3958261649302002	1.37230256946654	1.34748428660274	1.38723289518736	1.4359071619428299	1.34080490934713	1.3976050737112	1.3728781249118	1.34774609903657
BDMS	0.9552413656454459	0.961889385417861	0.8846252864505111	0.975620814856747	0.974202020492334	0.91496121215104	0.927433809495835	0.971520011380658	0.8836871750089671
BEC	1.7209715412503899	1.71288443163165	1.67461354817119	1.80762889985152	1.81414401945665	1.68383005509261	1.72263407025331	1.7388421785673898	1.6745726323571901
BH	1.4708440905434401	1.4812254647869	1.41700725709036	1.54338922369296	1.5428743724382001	1.48272580475171	1.48007434616045	1.4782212491594802	1.41713398981337
BLA	1.6972904375259699	1.71133395151379	1.64899948119679	1.6762845871149599	1.71223531115148	1.7240831374942402	1.70314958816607	1.71082150896133	1.64872690363977
BTS	0.934937113285099	0.9458803749847681	0.912107466175437	0.960174477736463	0.9457109850899851	0.9754601150510239	0.9573859800416961	0.9514827833668591	0.912170512506446
CBG	2.00181318848404	1.9916291055411601	1.9940339191486098	2.0507117859746598	2.03302913734008	2.06589628379865	2.03617873727385	1.99960872586139	1.99377400181171
CENTEL	1.50059829974063	1.5384829639072	1.48003925145257	1.5652802277650197	1.54796385693854	1.54346943361854	1.5023591494551398	1.5395699352329402	1.47982377020097
CK	1.42196056424431	1.4596103906870999	1.35418857894393	1.4241726620500101	1.4640343532378202	1.39240324947657	1.42641334072159	1.4986690370045601	1.35361529437848
CPALL	1.34979215270705	1.35968607261058	1.32615351426355	1.3430441474532402	1.3464635893075398	1.35057006113958	1.33997582284016	1.3386878146191499	1.3262250083793998
CPF	2.11818777345972	2.09868401703757	2.01811191284866	2.11827742581017	2.07806278079595	2.0298172271270403	2.1353464400692	2.09361726537791	2.0177705624485
CPN	1.2643371743582699	1.25232323535526	1.2282715748905	1.23854673055822	1.2559782468848601	1.2073215687268501	1.2500474193595001	1.25180329176326	1.2283130904469899
DELTA	1.7286997613900599	1.73836706981829	1.71234461683515	1.7904383764489	1.7600706115287201	1.73498288400241	1.73200376393841	1.7359300532118898	1.7121915215183199
DTAC	3.2050721373396795	3.19730542245325	3.07222129437974	3.17324117215798	3.14138467391589	3.17367456438787	3.18975623516127	3.20168084416523	3.07098278813259
EGCO	0.9587138890235529	0.966933356815667	0.9131825788611461	0.948951546362134	0.9642575390015401	0.9207519069964729	0.9592798270764541	0.970935914243622	0.913580760208917
GLOW	1.2783378300164099	1.31495135495966	1.2554992961824	1.32912552193854	1.3218707589172298	1.26303875783774	1.29091322342466	1.33184375106005	1.25549118754809
GPSC	1.48239551852996	1.51678560518878	1.48449570650694	1.79656244492332	1.7996143689668402	1.75510322612767	1.5762880861504898	1.54729848895126	1.4870464094676799
HMPRO	1.5303434968649898	1.52797689517456	1.46384937747108	1.50211122929458	1.5078595853604	1.45529571592699	1.53889100281932	1.5257203206052898	1.4628851766087299
INTUCH	1.6453554965406798	1.6566738467563	1.60382460555972	1.6626325896967002	1.6586273170072898	1.6163410103817302	1.64266355395422	1.6453864179685	1.6038282107231598
IRPC	1.34361250968129	1.38325800713936	1.28257287586247	1.3379785817927001	1.39257367649362	1.26600991738739	1.30402987099002	1.38276328478018	1.28269184830868
IVL	2.1256021567757197	2.11994388330737	2.08625594560484	2.2513974839327098	2.2542442818837602	2.15906813091356	2.12669628140295	2.11898839274932	2.08620333235242
KBANK	1.39094297515579	1.37607433642258	1.37612316315242	1.41738846331557	1.41432339484703	1.42058625648251	1.39464993022821	1.37620734385283	1.3762084511542398
KCE	1.53865196382273	1.5511374200068402	1.45505150185273	1.48514306737313	1.4845037010962001	1.47050035901503	1.52300289523609	1.5439758792724099	1.45521815453941
KTB	1.16642262973728	1.21561216915963	1.15482268874022	1.16610898590963	1.16572793059481	1.17943756072926	1.16932911025854	1.2152925552715599	1.1546367780173201
LH	1.25236447812447	1.22182579265661	1.2212913996855	1.24492765165571	1.2674169392230201	1.24707413161849	1.25535212835714	1.22159150551971	1.22132949443704
MINT	1.6038997058370301	1.6024995681419099	1.58391107959283	1.60049347608623	1.5973772154278298	1.62851231885071	1.60037046885196	1.60433874248375	1.58316190805721
MTLS	0.991406694952356	1.02427957876071	0.935463962563989	1.01849337496797	0.98876307615076	0.9991897796158351	0.983522285999206	1.0275120555194	0.9337242301904041
PS	1.37189173665978	1.42025050669955	1.24603036125872	1.35919889722206	1.4291866257889099	1.27450946978513	1.36942894732455	1.44228810069499	1.24599505772675
PTT	1.87674225711424	1.89151100789331	1.8317734757880197	1.8692638527919299	1.89081212236983	1.81773343515284	1.8734020404308702	1.90403411518121	1.8323009593662998
PTTEP	2.3277428803166798	2.31359939136477	2.23857504906174	2.20807135641139	2.19927258089227	2.1838988074666403	2.34446148391972	2.31069263569348	2.23869466145797
PTTGC	1.87193003982437	1.8646823595112698	1.7745125130242	1.85142058954635	1.85389326619313	1.86187855970737	1.90411604255361	1.8596682959898	1.77453964561276
ROBINS	1.5318045680643	1.57515884845137	1.46166905793171	1.5040610494909001	1.5327707267206498	1.46090997242223	1.5325584364523	1.57313045285538	1.46167125188103
SAWAD	1.49860648659856	1.5285362184098	1.39269783038942	1.52812836282168	1.5097416966828099	1.41048253847325	1.49793173963157	1.52756239449188	1.39228412835123
SCB	1.4705619922181299	1.50442775020686	1.47502778800432	1.47768840607216	1.45730323264352	1.45080289889832	1.4642282989548399	1.49933079695272	1.47501707923407
SCC	1.0889229970681802	1.11233344291306	1.07918083483177	1.09318886737428	1.1170739045793698	1.0748813241032802	1.09919205533727	1.14944446073694	1.07896775992242
TASCO	2.4727946251269	2.55076788447488	2.3288825584612	2.63433989781949	2.6892542421947	2.62991313873516	2.44716786322727	2.578947046269	2.32916155447904
TCAP	1.19679225919374	1.1747600783301	1.13087198890733	1.16070388045323	1.16640634745281	1.12768201186852	1.19784556861809	1.17770806180063	1.13055953027442
TMB	1.49283972318069	1.50172055312893	1.39068624878466	1.48006875550622	1.48522143690285	1.4762131786333998	1.47910914454953	1.50448271539397	1.39066418423034
TOP	1.52984836751893	1.52829203956684	1.49314003625742	1.53769389556007	1.58409954652461	1.51071099281879	1.5178451186925002	1.53266110218926	1.49360744178871
TPIPL	1.79669597275234	1.81089810898931	1.6260500862378802	1.91792141206194	1.9612098692221198	1.83128770163726	1.74252772519887	1.90796384993244	1.6261858368641802
TRUE	2.02618118190711	2.0233680619031102	1.9765333704724601	2.04239680930639	2.04143798617522	1.99543986903735	2.03990008287483	2.0222315680594	1.97652955276277
TTW	0.8171063197737181	0.862461027689879	0.808864816130984	0.888653811387412	0.894342012930566	0.827634466024892	0.812837990966443	0.858935540390497	0.808893365343977
TU	1.35338795929728	1.38017467812412	1.32847358591298	1.3899215892571501	1.37668133542243	1.33359480646452	1.3536354181111498	1.37886273255112	1.3281030274855001
WHA	1.5742678197229898	1.6292560833032699	1.51140258769543	1.6447674464561202	1.64322398313058	1.57565168232279	1.6010780745975701	1.62661026776811	1.51149857952694

PredictPrice5 : Buy-Sell Volume

Table 13 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	31.2515844205686	26.4913267182934	31.5573484820047	31.0385107317222	27.3955384055557	31.7578811360246	31.410358355358696	25.4615789382506	31.656504562224402
AOT	23.8207062835923	19.1093969467619	18.0361434424002	24.5387508165922	18.4257363282893	18.9381829775119	22.207621143503502	16.4029375896743	17.9746799256719
BA	1.05259065790219	1.05789090797927	1.01047762240529	1.05596364575264	1.03777693044974	1.04677810813545	1.03265653818313	1.10879517329794	1.00980998586177
BANPU	31.8668351878192	24.7916226711037	21.7587877546734	32.0114576522732	23.299249589735	23.844407224547602	28.445531641414	22.3938630124522	21.7965984060539
BBL	4.86071807205244	4.20686008191894	4.13805498799351	5.20837470692814	3.97082336808912	4.3049886198613	5.70072097300434	4.19697517721315	4.13207900644359
BCP	1.43776013670039	1.47594322203626	1.47299085616224	1.53402628281722	1.53655985724519	1.52470699786105	1.4312358939398	1.47367608562724	1.47190486756588
BDMS	10.0345041589927	8.83726338011634	7.837578190540941	10.7781273453677	7.9361236015267	8.19296379431568	9.98078509868421	7.91502072793088	7.79622522792512
BEC	27.574216003568804	19.083960634475	21.167921991234202	23.9274875142507	18.348385487516598	20.3391594826792	23.7034163019103	18.1984958173148	21.1855152800326
BH	1.82240210621842	1.75887037123531	2.3183086114006697	1.7599937538878299	1.64258978478603	2.66711686731029	1.93728188915957	1.79623848106027	2.32967956673354
BLA	13.0805322298392	10.7882828094431	9.33626576714569	14.704165437899999	9.92535817759527	9.63218740147135	11.8868693727101	9.282124296578221	9.30436481645015
BTS	1.16858587001653	1.10795928883692	1.16620520300227	1.19009808064562	1.1237845156117099	1.24973002980387	1.11663208980595	1.1020308110182	1.16538845221584
CBG	27.799965842275697	25.090537952557	27.516816463092802	28.2597903527383	25.4240644400334	26.749295288581	27.6163706391293	23.4203320860427	27.4500769187018
CENTEL	5.93668580917142	4.38133417025776	4.83844177077795	5.641963117965379	4.90902453133255	5.53049402916961	5.4268941715867	4.22422455240953	4.81559590068004
CK	3.1278468389723098	2.9064453676647397	2.75784605051408	2.8930403224717	2.65860594434089	2.83721837524278	2.89126784126381	2.7291407535861403	2.73239840930365
CPALL	2.1003379460086498	1.9722233900595	2.06060518840388	2.348469120322	2.0648952473577498	2.10220245698354	2.23873790657623	1.99928644528719	2.06081512914446
CPF	3.83583622887154	3.29060678626241	3.8781947918981095	4.24292134611201	3.4203888852840003	4.011416230013079	3.50488913274226	3.49530942224364	3.87652020136983
CPN	9.63896023238959	8.58660037308653	8.23228652864867	10.005925631082201	8.09967013076921	8.65033749837886	9.68372334481042	8.5375663437071	8.21577573142942
DELTA	1.91628676942413	1.9773609276450301	1.9948095137433501	1.88671198949884	1.97590482796051	2.0671179453855	1.87345701496295	1.9812266297755101	1.99457282534189
DTAC	114.711590613063	75.5834650798477	77.1527313726248	94.6039126192112	69.44749906286471	83.11660290782599	109.582910014164	80.0519397510925	77.1995157464266
EGCO	6.02592000927962	4.9652991997716205	4.781746751136811	6.56592717423698	4.6896597781167495	5.05491689492384	5.96739516614684	4.46111521926482	4.77916326441486
GLOW	1.33791624959882	1.40439694877459	1.4574237869067201	1.4024198131064198	1.45461167472909	1.47099339637077	1.34631776678762	1.39792512078702	1.45744511602032
GPSC	3.53245424719978	2.93697734188739	2.2592161524153402	4.077081455499821	6.6777995402924395	2.75230127886408	3.72764374877217	5.18801584659559	2.26927884212806
HMPRO	4.60672588334467	3.9310525866731902	4.23268730437628	4.93226437055654	3.98129814215765	4.47147423641296	4.34535101170679	3.99569577480017	4.23410698466182
INTUCH	27.51839720871	24.3072782597866	28.594801365217997	28.3218010930327	23.5569375672014	29.024441361681603	27.525180157100397	23.452886925258802	28.593887310910898
IRPC	7.5123243990990005	5.4039289109067505	6.37332574442315	7.46041497774344	6.5097605147893605	6.49493815410378	6.461516768369161	6.0546560768681905	6.38024683089729
IVL	7.29889549706138	6.4519263821430695	6.34879228123473	7.988671283311801	6.3380695631872195	6.9547031027017	7.51299578774485	5.90629234122623	6.338783370815521
KBANK	2.86709110567893	3.16119095337316	2.64545520953407	3.09162685370308	2.87096557979764	3.02873548139115	3.1518219463375603	3.23165981031111	2.6493574585144697
KCE	31.9763897395653	27.767489538333102	31.68408177343	32.3128598847195	22.368941230912203	31.7032888317218	30.5728431908177	24.4783290129645	31.626551374023197
KTB	1.78776110819395	1.88571839057424	1.75950714272815	2.10467963796661	1.91130340047204	1.90792174865912	1.8110959698056799	1.9715072661284	1.75899545969442
LH	1.49384135924528	1.41946657440086	1.53223683780683	1.41994134924633	1.39208531423319	1.5000158305616	1.48720616437362	1.40023692725466	1.53274555294026
MINT	14.0822659011353	11.390973235350002	11.1515810416395	14.753040225139099	11.071673335408299	12.1006319170013	14.4787060197073	13.1109648051008	11.0243159668997
MTLS	3.4131220362331502	2.8035946001636303	2.60690483683995	3.57501777354124	2.84235787937556	2.88817703958602	3.57036444764985	2.98759612771852	2.59866552688668
PS	1.36490089648396	1.4440875663926	1.45423666604129	1.4109383359843601	1.4167136170530499	1.58835045345437	1.35403278138768	1.42974706029245	1.4553490900289698
PTT	5.308925756312051	4.0118827036385	4.63477276875669	5.42899255602618	4.64645277053737	4.61524086226729	5.24639457006264	4.336106199581851	4.62689061152903
PTTEP	15.267639549771001	14.555584157999698	12.358327280164701	19.0778349619046	12.086304559508001	12.0602100750536	14.6526001030056	13.8663995655712	12.383791054252399
PTTGC	3.67623943143076	3.1372815531548803	3.53001419778433	3.75752329482397	3.3066228989471598	3.66133710914082	3.65744076126535	3.03993572705437	3.52965870870197
ROBINS	4.14839884254188	3.5839210152381997	4.0351278392947	4.96433915036494	3.9427438876040095	4.1252450747522795	4.21730755071126	3.9757239776642903	4.03102107252127
SAWAD	1.5595577877150202	1.44733795364678	1.38435085107553	1.46998956991209	1.37063598504359	1.39483183266515	1.5487278978762	1.45772183127594	1.38465924964199
SCB	6.61609107747678	5.71481264768014	6.2266249914759095	6.87568678905332	6.438419799286869	6.617462719091	6.98479535988858	5.68738475935369	6.2282563529112895
SCC	1.19607706098679	1.1534129775612099	1.1923514762704301	1.1306111924078601	1.1441941530365298	1.20822708754407	1.1979854242712	1.14663024903795	1.1931597500012
TASCO	5.243453238601091	4.23349921427662	5.85486988872918	5.03500076101865	3.8176145371365697	5.8214578250788	5.52971761397524	4.1822861117971	5.93518362419908
TCAP	2.58936853529608	1.9990957948642099	2.4797041614909	2.43486798982967	1.9844137291564	2.59399258452523	2.26625041315945	1.9699387309662	2.47279256807438
TMB	1.7057457081964902	1.7023227529412301	1.6530714495516698	1.59671898910846	1.75629684819451	1.67731883167015	1.59445623553299	1.7120793604312599	1.6533830718328102
TOP	18.2302471895709	14.725395134636601	15.8275502199206	18.0656780506227	13.7915037491846	15.5293232683655	17.297361623285802	14.0771063986892	15.751149899192802
TPIPL	2.20866875530611	2.29661183967601	2.5188949989296003	2.27901135872881	2.24629511133472	2.5007384115568	2.15732979434788	2.3056412523401604	2.5239940322253798
TRUE	2.44268451056403	2.41795175279546	3.37200709818659	2.4403429894660302	2.3733623950486002	3.47994892692274	2.5896779443188502	2.3861929931067	3.3658512875497695
TTW	0.944914934690324	0.996075517145168	1.15373907021196	1.03667994077483	1.00415718774824	1.29509762115588	0.951545770324874	0.989662280941059	1.16124152492528
TU	1.38809235874345	1.38202070114442	1.42787256796725	1.41339549557014	1.35842655544053	1.46890571083798	1.38635749473075	1.35203100439513	1.42932829121767
WHA	1.7450800099302202	1.7261453636770299	1.76650253552386	1.8623074504134098	1.7717343859057402	1.99870498245097	1.70782984764472	1.7352197986965099	1.7659994350103398

Table 14 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.80589483288926990	1.81759710099794	1.74619497773798402	1.80556936306955	1.78190543871	1.74613924562554	1.80490378800424	1.78549656094549	1.74598775969472
AOT	1.06187788390794010	1.04090761761240010	1.03897531530511	1.06666974043603	1.04728660226601010	1.04967411686423	1.05972549367618	1.04119658675498	1.03973234948921498
BA	0.90608577501269800	0.90141127527513800	0.92243101429100490	0.98629465059752400	0.92457555105543300	0.92002404177499810	0.93839962464738400	0.90116252986530700	0.92337890514741690
BANPU	2.03351966783347	2.05786210888865	1.99102520603199002	2.03212734500082040	1.9960461085142	1.97515064042203030	2.03432203358733	2.07602774019685	1.99101965445248990
BBL	1.25215176673698	1.24985098516427980	1.217633868818250	1.22963319659597	1.23257685152665	1.23063068581728	1.23360939622722	1.24809657835725	1.21775363088338
BCP	1.38875313253383	1.37725138684024980	1.35147813198932	1.3873617922749	1.38956240394850020	1.36586647956197	1.38629275979476	1.37342857836329	1.35157563335240980
BDMS	0.95463302792619910	0.94700290562763890	0.91126677349408910	0.97796082497063710	0.95195446391413910	0.92476854857426300	0.94994977933329400	0.94542488679955540	0.91078834680647400
BEC	1.75939562457952010	1.81791802213858020	1.67608155198116980	1.77986503264929970	1.76297568427116	1.69748829520793	1.77220207650047	1.78302050916884980	1.67628904905793990
BH	1.44565723058460980	1.44415220559710990	1.41470885847209020	1.55211273515791	1.49870322963254	1.47889273065265	1.44890957560490	1.43980858041167	1.41451502579046
BLA	1.71043421071611001	1.66414146854321	1.62022599739084	1.74082473061277020	1.73925062738497	1.7443524826911	1.70902711070119	1.66483733354289	1.62002314312724980
BTS	0.96565007792230100	0.95495467244338590	0.93056173924167300	0.98296603338628250	0.97028513434058690	0.96345529373715700	0.96722752548138790	0.95579785697816	0.93014561036807200
CBG	2.08372369579986	2.03447853793513020	1.98111716977020	2.13887096217529	2.11497792933017	2.15216146111563	2.03063908695701	2.03427765906399	1.98318692259078990
CENTEL	1.52500627365133	1.53209029613289	1.5157734253495	1.55555701074122499	1.53305125697521020	1.55222719092394	1.53300287916398	1.53220370937046	1.51581826824538
CK	1.37656063058062	1.38642677663348010	1.351081008827	1.33395159384013	1.35507459099169990	1.34691381747067	1.37814159417447	1.39192545186363	1.35114300348953
CPALL	1.36728996842888	1.37403946071437	1.33551473082485990	1.39755737545254	1.37683316629112	1.36351267508701	1.36482206139934	1.37138599500276	1.33517780647776
CPF	2.03513508256546	2.0526852946742	1.99869210187739	2.04950470666359	2.03436205120575	2.04883827508784	2.042197935727	2.06005768226188	1.99880509955000998
CPN	2.24929242229397	2.24160074874194	2.23529958072011	2.21734038117908	2.21278210400722	2.2104274967944	2.24545558612323	2.23920006960984601	2.23530429187387
DELTA	1.83891582336362	1.81549986989950301	1.748520681588590	1.83858486842878	1.76636128065719020	1.74404184158075	1.83804141579984	1.81700019422296	1.74838851644462
DTAC	3.15117939196259	3.16224465018025	3.10066298218471030	3.26123274094823970	3.21548553888646405	3.12339877526103040	3.10782771550596980	3.1822409339978	3.10021586360058
EGCO	0.94105482431521590	0.91684346822032390	0.91819463193652410	0.92480157111542591	0.90532455771024110	0.90586377192317400	0.93919236723496600	0.91650739614526760	0.91830145941684290
GLOW	1.38515808342833	1.35420436553292990	1.27090894891333	1.28164215188023	1.29580005312516010	1.30724307691585	1.35622988554201	1.35731966530399	1.27104302810668
GPSC	1.55290828250989	1.55089460419572010	1.61122268747042980	1.76342912494503	1.78948651822775020	1.71803832182333	1.51381782101322	1.53632254109274	1.62340784780758
HMPRO	1.53091452490988010	1.5338402556241	1.4960304715464	1.50043640704506	1.4832080260225	1.47881665524048	1.54318992070542	1.53874717494591980	1.49609217233382990
INTUCH	1.64253110759176	1.65500647082287020	1.62025220731403	1.71549948330428980	1.69106709556549980	1.64948904009785	1.63877009535554	1.64948611169103	1.61976015935011
IRPC	1.30435773765001020	1.34638385610541	1.2879841108524	1.30860563450271	1.32944333658628	1.3158733892145	1.304137782498	1.33365513392117	1.28791252234752
IVL	2.18926160159871	2.23357664866005397	2.11400453490162030	2.30866530444409	2.25848482519539	2.16169966585826	2.19235251871714	2.2236246698456	2.11419196552121
KBANK	1.40285634525174	1.38116923517273	1.3858503741617	1.42072000058094	1.38802656926277	1.38701225246528010	1.39758426349015	1.38111327733569	1.38585609556605
KCE	1.51211134808475990	1.50022980988078	1.45697998643184	1.48878612226376	1.48104287427047	1.4691963344444	1.51202395092925	1.49052269202942990	1.45692082887405
KTB	1.21572902767754	1.20459380768878	1.161224876779996	1.21238816015543	1.19844879197982	1.1782243694183	1.21568562728558	1.20499850408397010	1.16076024118612
LH	1.29221952745365990	1.29296089466409	1.23081385195111	1.26368525780907	1.27284191892706990	1.20304321371247	1.29104333803894	1.29960110915109	1.2308075416433
MINT	1.67222688207423990	1.64193109668862	1.61665428233443202	1.64596530792013	1.61286659954867	1.61608971234889	1.66389118883052	1.63694075290282	1.61710532931027
MTLS	0.97497533262668	0.96007628290882510	0.97031861666248	0.96344767586484819	0.93864782256553	0.99027038190504 01	0.96607555533080 29	0.96987267547942 2	0.97195111428066 1
PS	1.36369534258611	1.32850702096302	1.2699410105138	1.29397773769687	1.28648558543571	1.2775811599148 2	1.35299074778784	1.33661390967799 99	1.27024679053389 99
PTT	1.87890254467553	1.88118362982113020	1.86360039845158	1.89122491331620 02	1.85222875979697	1.81257554042471 01	1.87774865235883	1.87881345646003 98	1.86328680961664 02
PTTEP	2.50312129443424	2.55005888334563040	2.32678112166801	2.30817670703347	2.25609988974522	2.1864909417167	2.54277954732857	2.55309512429615	2.32668824391282
PTTGC	1.85850949219388	1.8330434791202	1.78966444572733980	1.87578395534133	1.88392425981059	1.85030728674892	1.85516787740569 01	1.85537146966332 99	1.78961087958252
ROBINS	1.53938790301092	1.56780375193937 98	1.46889715315859 02	1.46961846753347	1.47251280475374	1.47413580852793 01	1.54994103632759	1.57003572373294 03	1.46885901556117
SAWAD	1.47297444717611	1.45006028736036	1.42335795380448	1.45141862090461	1.44477276719809 01	1.41915976090496	1.52103289938275	1.46012576751030 99	1.42301491737861
SCB	1.47744616245708	1.49334636874359 02	1.44108983246970 01	1.45668625379492 01	1.44837238626718	1.42471630353291	1.48216120328466	1.51532875610671 01	1.44114099701832
SCC	1.08102671773778	1.06323708181188 01	1.07521889746353 02	1.08288365978302	1.08799458703842	1.07381994503592	1.08054754484573	1.06264415932633	1.07509795514249
TASCO	2.38756268673344	2.3967999210815	2.35008009447122	2.50132613662142	2.48154135786594	2.47369970221758	2.42775984361504	2.40120266821312	2.34831200584323
TCAP	1.17578818060933	1.16251131437844	1.14106592644406	1.16525972510517	1.15507891086099	1.13590096940726	1.17709789234364	1.16741714454054 01	1.14099243233149
TMB	1.45973929690078	1.45180657187424 01	1.41610122091605	1.4493148317602	1.45458302632374	1.46087651586924	1.4644035131278	1.45313066141914	1.41593550508 06
TOP	1.52293285856399	1.51855814157742	1.49217566932703	1.54316510580849	1.52712492429669	1.49636296874945	1.51801517688605 01	1.51092940746839	1.49219675541249
TPIPL	1.65896214240047 01	1.70587710613624	1.62119961277665 01	1.94365765649078 02	1.92833277105942	1.77142490922777	1.67828799378900 98	1.70887981590095	1.61991850913454 98
TRUE	2.09849577448795	2.04931839871018	2.04212462786833	2.14710200886468	2.11722620657870697	2.16368311738813	2.07204477018918	2.05957756844246	2.04194909035874
TTW	0.81298070198113 33	0.82955889779317	0.80870883410604 8	0.86358297660311 2	0.84232452741445 8	0.83976740644347 5	0.81465453960747 1	0.82038537105339 3	0.80869031882523
TU	1.35121760837061	1.32059484952529	1.32370713320509	1.30226967539210 01	1.29612255339	1.30914263200524	1.34694339099784	1.32073872558733	1.32397157396530 53
WHA	1.58823777644397	1.59217738027774	1.56633441333296	1.67622529074354	1.67011821646665 02	1.64173482989075 01	1.58803783390287 98	1.59813772066232	1.56620777002469

Table 15 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.74998239220271 02	1.74794759508115	1.74559849560057	1.77095858278468	1.76306503608527	1.73852056850779 98	1.74813943045092	1.74622501076092	1.74554576888146
AOT	1.06327101097105	1.06935865540477	1.05024806952779	1.06304541638342	1.02971590349856	1.02980071910318	1.07400640192261	1.06735562153766	1.04955211848392
BA	0.97149950948415 9	0.91108138700493 19	0.89083955762211 21	1.04231305407123	0.98000194719947 11	0.91942331210695 39	0.94482397396553 3	0.90262066939180 21	0.89088318538646 39
BANPU	2.03766834174148	2.09885604252815	1.99236449867225	2.04042158545301	2.0050617347906	1.97927158550386	2.03560923084999	2.07181751427654	1.99229864514798 03
BBL	1.22497932839132	1.23047911194626	1.21486478157744	1.21829528921796	1.22791463914576	1.22270130658403	1.22205517159787	1.23096767151638	1.21499418350316 01
BCP	1.38178843434644	1.37508892906819	1.35147813198930 99	1.38642773855456	1.38932667152851	1.36586647956197	1.38889008418261	1.37439332737744	1.35157563335240 98
BDMS	0.95739438950703 9	0.95090885025856 79	0.91127150425622 8	0.97555352312561 69	0.95966119195975 91	0.92476854857426 2	0.95036138700601 9	0.94312110678849 7	0.91078004024623 381
BEC	1.75826126708525	1.75248284405244	1.67195563092333 99	1.74791764603066	1.73538173322798 02	1.68745055483326	1.75253593835429	1.74888143585478 02	1.67227348252245 99
BH	1.47766903467022	1.47811760927126	1.42056546609219	1.58517204984922 99	1.54700517880836	1.50259829280168 99	1.49016551102 37	1.46342746369829	1.42071977282614
BLA	1.70317681155103	1.66110218185430 1	1.62018091410452	1.74979741080905 02	1.73976495069753	1.74434501705044 02	1.70126683955883	1.66371019591886	1.62002002005769
BTS	0.95305940573244 4	0.94585773872241 31	0.93068206572837 3	0.97052069308458 01	0.95610741540863 3	0.96654541381 68	0.94897685742344 1	0.94540904762766 5	0.93026596312260 4
CBG	2.01598059980285	2.03320458116258 03	1.98111716977020	2.12912949424 78	2.10549253569548	2.15216146111562	2.03254328354257	2.03779507097884	1.98318692259078 99
CENTEL	1.5268573521394	1.54249219516920 99	1.51434019624782	1.55598921637986	1.53491040505024	1.54991273764707 98	1.52928593617537 01	1.53970181029602	1.51425141152649
CK	1.38096145006153 01	1.41496238526433	1.36269641056593	1.42439434637 58	1.40460294680614	1.37674822326179	1.38496798288864	1.41386033484041 001	1.36126631484056 399
CPALL	1.3685295413264	1.37292147797748	1.33514730824859 9	1.39597670132682 98	1.37377528447514	1.36351267508702 01	1.37470985990983	1.37759779337009	1.33517780647776
CPF	2.07775679656387 97	2.07263334039212	2.01713487495604	2.11047923126095	2.06411866030587	2.04956902730198	2.07677725093686	2.05572839464450 97	2.01690572954397
CPN	2.24820834063242	2.24100435573804	2.23529958072011	2.21763586768490	2.21457587711736798	2.2104274967942	2.23593351304233	2.24241685374771	2.23530429187387
DELTA	1.83171018478263	1.81089560656224	1.748520681588590	1.83301546940096 02	1.77736022960198	1.74404184158075	1.81913174429985	1.82096807486134	1.74838851644462
DTAC	3.13641622580942	3.18662923404171 97	3.10573585523808	3.16768901417803	3.13524968523998	3.13147638134856 98	3.15676490941627	3.18438271766395	3.10621899877293 03
EGCO	0.94037064429111 39	0.91838818583238 8	0.91831028027468 1	0.92932266989822 31	0.90733932470552 9	0.90642724283954	0.94043183900345 31	0.91880789669230 2	0.91841619412986 9
GLOW	1.39866159961290 1	1.35188133346174 99	1.27648264990305	1.28384397648855	1.30290160474164	1.30822540712085 01	1.35530092858829	1.35184554934567	1.27643843010792
GPSC	1.60060707733275	1.54841606369298	1.61122268747042 98	1.73734444432981 98	1.78302869759882	1.71803832182333	1.51099402650619	1.52672648705336 99	1.62340784780756
HMPRO	1.52820207426148	1.53083719304456 01	1.49468703563047	1.50458215389697	1.48094245868233	1.47715876468678	1.54371202986 38	1.53935349233906	1.49488348787282 702
INTUCH	1.64142734029499	1.65500786306313	1.62025220731403	1.71432545552686 99	1.68339926395564 99	1.64948904009785	1.65135886002391	1.65464257394839 99	1.61976015935011
IRPC	1.32135598542119 9	1.31541811506127 98	1.28387716965255	1.34180867695191	1.3329572900145	1.30652734935652	1.32438982745 52	1.32944314013339 02	1.28368014224635
IVL	2.17976560108441	2.18761189040683	2.08304580950347	2.26261326947146	2.24682912980037	2.16936349085174	2.12888221340505	2.1899708300382	2.08227821884321 96
KBANK	1.3978985811755	1.36522636893158	1.38849246036917	1.41265547906428	1.37921229605195	1.42138291127564	1.39912638622309	1.36451475086029	1.38904896847295
KCE	1.50145767061613	1.49804796508079	1.45265582660007	1.48260405994015	1.48397922910745	1.46140921630821	1.52034178217643	1.49465785659448	1.45310626084876
KTB	1.23742150880168	1.20234412889692 99	1.16119688202699	1.19842982800107 01	1.17822436941881 01	1.20290058354568	1.21245695606786	1.16083009157955	1.16083009157955
LH	1.29112770519912	1.28168399896271	1.23594521925449	1.26360060970572	1.26213338135781	1.24046640578684	1.27038167491139	1.28061344771801	1.23590949746158
MINT	1.68095619835399	1.66409328311522	1.62045979878649	1.6744742343666	1.63966541369868 01	1.65107968217436	1.69312967100403	1.65160213179509	1.62063307789839
MTLS	0.97537995242029 91	0.97164729911946 2	0.97031861666248 21	0.96430803032256 31	0.94331727854041 5	0.99027038190504 06	0.97242802717782 6	0.97010026537891 9	0.97195408819335 9
PS	1.33207540876076	1.42959966542719	1.29465739827581	1.2622189156084	1.30267137444441	1.23534733064864	1.41548396316448	1.40174492372 72	1.29498073898714
PTT	1.88308001696943	1.87577416301679	1.86627602906956	1.89508680741324	1.86505364970976	1.81493162744480 97	1.87839776427937	1.87686387446576	1.86595602809011 98
PTTEP	2.43547281539319	2.40150401159694	2.26870900165623	2.27261964321003	2.22048412238318	2.21167463778207	2.46138871740025 02	2.38556097182247 97	2.26885891960446
PTTGC	1.84536392470628	1.84579473321665 98	1.78966444572733 98	1.87524723794358	1.88256485901927	1.85030728674892	1.84775960317446 99	1.83269807707334	1.78961087958252
ROBINS	1.53997365064371	1.56192349413108	1.46889715315859 02	1.47585429077141	1.4684378824222	1.47413580852794	1.54854956559438	1.5638432323347	1.46885901556117
SAWAD	1.54193960766129 01	1.45495974809029	1.42335795380448	1.46288673045282	1.45108175819 08	1.41915976090494	1.48706295769308	1.45037654974540 2	1.42301410554674
SCB	1.49220441023663	1.48993028398559	1.44108983246970 01	1.44615493146448 02	1.44254734000779 02	1.42471630353291	1.48233622974293	1.49170339683064	1.44114099701832
SCC	1.07704937149604	1.06277956711736	1.08023273403297	1.10835071105779 01	1.0866957994688	1.07876936670122	1.07951084283813 98	1.06290279939869	1.08029929187684 02
TASCO	2.46583958638971	2.4587542264989	2.37082659298307 96	2.71226648725338	2.71069975713828	2.70104881348834	2.46883119068703	2.48830010907602	2.37110934498561
TCAP	1.17629309326625	1.16547775149511	1.14106592644406	1.17628898311987	1.15251559312772	1.13590096940726	1.17945397920979	1.16353910148244	1.14099243233149
TMB	1.46176835061152	1.44827629784125	1.41643778429712	1.45340579192794	1.45566168260103	1.45864198979376 01	1.45990011394967	1.45116933648545 49	1.41639072140 6
TOP	1.52064598877966	1.50537051953432 02	1.49217566932703	1.54520406408102 98	1.51872844712 52	1.49636296874945	1.52368747854169	1.51711203192349	1.49219675541249
TPIPL	1.67699842895398	1.69871746436133 99	1.62787247695974 02	1.92644177286423	1.96774122583919	1.76665183140143	1.67602933283672	1.68871744846911	1.62817647460167
TRUE	2.09457771995485	2.04511195772407	2.01538400672869	2.10343010677262	2.06583678808272	2.06496613634560 02	2.10555188051180 92	2.04393462348941	2.01529388247909
TTW	0.81216519670035 99	0.81176607283874 7	0.81085712946908 89	0.85057260934461	0.83508767642972 19	0.85458720366193 9	0.81230939481510 5	0.81447112271589 79	0.81068664882047 9
TU	1.35259441848449	1.32054479637137	1.32372713043951	1.31446330542054	1.29322863434749 90	1.30914263200523	1.34810969363355	1.32080514835115	1.32399246085608
WHA	1.57853894158201 01	1.60294202763502	1.56411937109718	1.64687287203923	1.66239170059 34	1.62824140939611 02	1.58297251589406	1.59383609940051	1.56405969654796 98

PredictPrice6 : Buy-Sell Volume with ATO/ATC

Table 16 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	29.9975651734183	22.1497722342119 02	35.435421645741	30.3159881233138 97	22.3735705737132	34.7217186638165 96	29.4004863781378 03	21.174897115876	35.4552338754827
AOT	18.008097878294	13.8261268363 71	22.3304058911053	22.0451435545405	13.1066704079227 99	21.6233314312503	18.5827014981422	13.9537444205353	22.2714428483213
BA	1.2465157145174302	1.1367200286354 2	1.136903992005701	1.162364693413 05	1.139408242571 03	1.1246144108947	1.1743724073889599	1.137401949675 36	1.135863873764487
BANPU	23.241896555441	17.3441534050030 98	28.211372467976396	30.7905789960619	17.1480059698712	26.7227098201342	27.8318010989151	17.1257193085061	28.2308206622626 04
BBL	4.4838098459705105	2.961461708111 77	4.7282211233639 7	4.753905603515 25	3.1727289233751004	4.846861468628 93	3.855267954362582	3.1715840774288804	4.72586027003457
BCP	1.3863359626774299	1.366370090617169	1.519356040687 72	1.41482906582813	1.4015600601275	1.521615427272 22	1.360142849287 44	1.361234050860 01	1.5217100498277398
BDMS	8.451347127667031	6.114100818319 35	10.061974206251099	8.206177300796421	6.1529745087357	10.684844134541 5	8.880110073623 49	6.1504583089408005	10.039397474441101
BEC	18.57911342565 54	14.642674799513 1	29.247045984007098	22.3622189043691	13.9142576133979	25.617668570774 2	19.589166415315 6	14.7798499037823	29.2419990327421
BH	2.309530053524 43	1.788444857224 08	3.235631904799 03	2.312583282434 14	1.745908303033 59	3.142885526824 667	2.3222682798942103	1.8424394157883	3.23134091103121
BLA	10.038043514307 1	7.670749651990 54	11.1428575506324	10.5776319902	7.9157146163428695	11.720620562249 9	9.676113163742 3	7.83186148743429	11.1431794365652
BTS	1.1263010324865799	1.076505948527 87	1.304596089223 95	1.200650442242929 8	1.1044103796538	1.293464354834270 1	1.140287854712 91	1.0723518792451199	1.305501895993 11
CBG	26.589436806 8834	22.823867237268498	27.299377957347897	26.817341878903	22.2965570415269	27.235222360390598	26.712945333445 7	22.8881757100908	27.3385207965654
CENTEL	4.80627396711192	3.922026590476	5.327797870606021	5.302334054409 75	4.1649397977116495	5.974940423246 83	5.048306581867 47	3.7414133874560 68	5.333366052170 8
CK	2.6177940068777503	2.528197233838843	3.188764474242890 2	2.841535443801 83	2.383421381748 91	3.250389743344 08	2.680625740442 112	2.4377660404449 47	3.1848309000425297
CPALL	2.008408164874 89	1.884689195910098	2.334135980173 28	2.247087347721 87	1.9278625110068701	2.325422836708 11	2.108997693314 19	1.876462511077 07	2.333805256404 71
CPF	3.1860533515394502	3.017467873791 57	4.98934690375312	3.874950870311 59	3.1781731995990103	4.740081785850 5695	3.374801397806 25	3.1115301823199 6	4.998256117977809
CPN	9.078342143233 028	6.331541876559 955	9.798299034089 97	9.432211193157 11	7.175884434312151	9.471889852028 72	9.344781698187 61	6.3408903921454804	9.783779228559 4
DELTA	1.939989728183 41	1.901057745775499	2.222612687203 93	1.996129672919 44	1.909966094610 38	2.122324589480 46	1.938755795055 34	1.9121659207516202	2.2215312746275697
DTAC	84.524065062618 5	57.013560040030 9	108.0090422484 94	86.31942577918 79	55.196322701947	104.81164915645 7	84.629890527753 7	61.097479787138	108.045964776325
EGCO	5.717839284767 38	3.677105272233 428	6.260196758167 02	5.210262547987 15	3.908968727202480 2	6.227549503216605	4.806915521217495	3.5824835548539804	6.259862255203889
GLOW	1.380769677078 79	1.349369874418 87	1.698729481209	1.433080980077 58	1.373381047404 4	1.707137679241 87	1.415900982275 42	1.366575990523 84	1.6956237830688898
GPSC	2.706607684803 82	3.226621731947 04	2.174259474133 7033	3.608863034678 55	4.56191534393659	3.172516156836 75	3.620596074067 0498	3.6651868029942 998	2.176386949855 27
HMPRO	4.30658067396 06	3.484380429650 1304	5.006915327464 8	4.588001893513 64	3.513960113994 6604	4.920494117949 78	4.362066333287 11	3.3678700775 6325	5.011349379333 7805
INTUCH	27.246857186896	20.798163486105 402	32.85391514058 82	27.216715250477 503	21.47546090032 48	31.638473967093 104	27.577364122120 8	20.932660843 70 79	32.8553351642653
IRPC	6.1018022622002 5	4.744362126928 28	7.570602808858121	6.98233342141 6	4.806891295678279	7.313122338042 35	6.233023845824389	4.6544870147926 2	7.573280854290861
IVL	6.369606621764 26	5.131667040540 16	7.276408112773 55	6.600960399343821	5.280394951962 57	6.839431089539009	6.166901479101 16	5.1304940790834 9	7.279652203452095
KBANK	2.95836789566 54	2.376621802647 1297	3.463078881306 24	3.01568561479 36	2.523139777167 24	3.561588584768 18	3.160574461860 56	2.3571817824 3588	3.46259124631052
KCE	27.842690834797 7	20.126585504822	40.024338594315 9	29.055859908 1993	19.214348033449	37.210397823123 8	29.601942166218 297	18.6448881958 21	40.007746956952
KTB	1.854078936162 44	1.820707302135 7	2.168466373462 77	2.010978615876 14	1.834892881237519 8	2.476196663917 49	1.856026148480 78	1.8185930529030898	2.166532864127 73
LH	1.436477921216 58	1.350759039125399 9	1.752658631540 57	1.378547536623709 9	1.310739627038 44	1.673405572798470 2	1.438602838630 432	1.3600536980432	1.752149059985 62
MINT	10.746099320565 9	7.976094148392771	13.877431948607398	11.831033732036 1	7.52525867012674	13.198706758673 1	10.939482367500 8	7.5071562678874 51	13.868501296904201
MTLS	2.925336276642 89	2.456922564858 85	2.706778161906379 8	3.092317347993 03	2.679359410622197	2.969971314708 73	3.189942465482976	2.4480911605060 49	2.696596302167119 8
PS	1.423001399248 11	1.463478129007 7699	1.696591836943 375	1.509510887789120 1	1.450405333819 17	1.794759862608 4	1.428574857731700 1	1.4618195410079 5	1.695615333657210 2
PTT	5.265972684737 74	3.595985058915 1	5.376943675081 27	5.033043397615 49	3.851404088942 71	5.216203537354605	4.365305789004 39	3.6685668044 5439	5.382319573986 04
PTTEP	11.489980277805 6	10.887060702940 5	15.265806234453699	12.1856906073959	9.955876146221 35	14.069986103787 299	13.191754398190 1	11.326352611 727	15.295213276111 3
PTTGC	3.213039294767 18	2.808388863190 04	4.085604493777 08	3.391723190634 04	2.871847853972 73	4.107304434839 58	3.193521267302930 5	2.8117739529184 9	4.087934486652140 5
ROBINS	3.718939294189 38	3.083416147984 14	5.184066474023599 5	4.089412653911 1895	3.101491829200 18	5.076773553978 79	3.814223134709 5303	3.0193878343115 04	5.181870681290 74
SAWAD	1.540678582035 21	1.500937640302 198	1.425943928100 2	1.629957223684 9298	1.515522506095 4	1.546772189789 13	1.562703776097 01	1.4693131791 8977	1.426309485379
SCB	6.384835503546 76	4.861336279361 73	8.490975642821539	6.278431505730809 5	5.007647355388 4	8.102544182284491	6.355198676074069	4.9749322991060 1	8.485155788243 99
SCC	1.156843453552 47	1.173071245534360 2	1.295152183632607	1.219650256189 3399	1.164915923986 97	1.274352360748 5299	1.159644018488 21	1.172393122 21	1.295459446773 85
TASCO	5.016107144281659	4.630722326390 67	10.0039841659544	6.692944945380 17	4.247732164972 64	8.533569270071869	5.627082705744 92	4.5733142712424	10.047008360931601
TCAP	2.417806977929 55	1.848946089048009 8	3.187964472472896	2.463401990629 21	1.974067959768 13	3.041984387885 55	2.220706650177459 7	1.8481970736813	3.185725709617 93
TMB	1.580609140805 65	1.611880371416 32	1.731775003894 86	1.622108509803 34	1.635444549999302	1.952315547233 86	1.571945274816 55	1.5838369595526602	1.732806416096599
TOP	15.120545412168	12.094526160559 1	17.505749378020997	15.249229690850798	11.689632016528199	16.634483977356	15.565213704499 1	10.826789551516	17.501283961942 9
TPIPL	2.172127603662798	2.182034879461159 7	2.935877289651 33	2.244942140073 53	2.207478018200139 7	2.749315844726879 7	2.293842665371 12	2.222058712511 9	2.935835821683 64
TRUE	2.860116951542 15	2.639778327316 82	4.173837363276 17	3.003967827897 62	2.552215596329 44	4.365303446483 86	2.884033588297 9203	2.628554675951 77	4.183968152671 03
TTW	0.965788206317045	0.981526659163071	1.269990973501 97	1.269885941694 08	1.026599248887 71	1.377670251624179 9	0.957225963506835	0.991171930214164 9	1.272206832249 43
TU	1.527881171738 98	1.498548086334 11	1.707642621357 33	1.556046464462 23	1.489111241317 2	1.675608192832 68	1.587985472332999 8	1.5013312652759399	1.708518856342 6
WHA	1.677414720657 55	1.717318181801798	2.102429348680370 2	1.919886772206 22	1.826168837567 17	2.191337276403370 3	1.722456563569 78	1.741102272665 75	2.100349140725 03

Table 17 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.77703131193903	1.765092492127 42	1.756884174594550 3	1.800113406597 52	1.763539201418 8	1.770482537440 44	1.770131795209169 7	1.774500354977 3701	1.756840442905 75
AOT	1.054286940729899	1.040470653293 59	1.042774765861 25	1.062794645580 74	1.046796151883 32	1.056063488214 897	1.059744196516 44	1.040063057245 4101	1.042881364223 96
BA	1.0158768488833	0.960882919257099	0.983326912366857 9	0.921955244712066	0.904135644004942	0.921716459226129	1.031877193980 7	0.956954770668489 1	0.980884634887129
BANPU	2.07202420181178	2.045792883896697	2.008862154826 28	2.032581223924 89	1.989507475863340 2	1.978015462374 14	2.062996221404 28	2.055393409862 1	2.008240197746 07
BBL	1.222746113653 89	1.220048652311 12	1.225446903382 47	1.213226946979 05	1.231692555331 23	1.236589468154 47	1.222030302530 81	1.218694515565 14	1.225385672517 04
BCP	1.372075980610779 8	1.352157381706070 1	1.350955217737 91	1.417500849297 62	1.384335017644 72	1.402530858140 9202	1.372121708331 1	1.351573245823 43	1.351041673668 43
BDMS	0.911389564050024	0.912475203986486	0.906957294072432	0.934675790866 296	0.926719572997567 9	0.923041949963680 9	0.910670374183761	0.914518753269055 1	0.906894501033607 9
BEC	1.818900573044 63	1.794775570114 72	1.724455657152 15	1.849783228566 48	1.788545238078 1102	1.792301668318 24	1.824363544367 48	1.790217940373 4901	1.724561822652 4502
BH	1.41947774635 1	1.415539697280 39	1.433951739089 38	1.498721978343 89	1.470449424254 79	1.490064924878 81	1.418652203618 96	1.416404224803 23	1.434127375036 28
BLA	1.659387016321009 8	1.621149827732060 1	1.616298772407 83	1.765897888393406 2	1.746509963513 43	1.764356315033909 9	1.658986155648 75	1.620765265674 28	1.616321865066 05
BTS	0.963934907112144 9	0.946720177299 2	0.951545128430654	0.947505428718790 1	0.937651991726214 9	0.958481029133742	0.967140258251696 1	0.947580412918 281	0.950773698872301 1
CBG	2.091631498635 52	2.047181600119 65	2.024195780969699 8	2.174682763853 4	2.108389311582 08	2.184740211002 7	2.093110334382 84	2.051250464282839 7	2.025008411460589 7
CENTEL	1.519669968203 41	1.495190559628 78	1.526649504708 61	1.554834459260 02	1.540125103493 85	1.563229584025 13	1.517844612665 34	1.499263740850 4	1.526707743696 73
CK	1.361935833898 9	1.350766934321 368	1.353177743513 17	1.340283721272 87	1.331852298449 91	1.348420509582 12	1.364016319954 06	1.347760158277 33	1.353809937506 68
CPALL	1.395590001217389 9	1.358291469115 4	1.351571837979 49	1.391165482850 03	1.360729919481 35	1.359184967717 21	1.396055732047 3401	1.356550641705 4	1.351567761521 6402
CPF	2.034769241934 15	2.024591082526 69	2.000326050222 504	2.057997912874 35	2.018899031194 04	2.034226022620 83	2.043362357942 34	2.026857454500 47	2.000098534216079 7
CPN	1.277869696302 27	1.275232003588 28	1.265204255268 94	1.253618661582 84	1.236827918284249 9	1.251906552446 78	1.289418105409 46	1.275022802459 68	1.265290118464 24
DELTA	1.823101275877 91	1.813935207294 1002	1.784546344487 6202	1.786538071309	1.756395954879 06	1.760318702982 4299	1.835691640952 1602	1.795673860671 99	1.784513052212 7401
DTAC	3.158025196823 5005	3.208058350949 59	3.105064557573 17	3.079538635672 23	3.100915543880 9	3.132661368847 3	3.163708436295 697	3.190718054714 605	3.104388096723 62
EGCO	0.923908989957614	0.914479751938 123	0.913521187335484	0.928381519956 85	0.914547073450676	0.927899997289274	0.921428184327608 1	0.916060406502153	0.913585806887 95
GLOW	1.336982611245 45	1.327218859707 857	1.281921508057 96	1.309150103261 64	1.319754650655 26	1.312755011289 61	1.352291998959 09	1.333041273969 57	1.281384917606 12
GPSC	1.557407538211070 1	1.488146195225 42	1.662099457351 46	1.771617308315 49	1.793693086803 71	1.705508626386 9998	1.476289872243 43	1.498604306352 2299	1.680158178131409 9
HMPRO	1.494367958120 91	1.500300027559 8	1.496608018333 19	1.499468147223 9	1.490963696012 57	1.478784235111 83	1.499645324815 1	1.503645705627 49	1.496446525017 11
INTUCH	1.66840336288 74	1.665469641670400 2	1.668120679840699	1.731241093733 22	1.700410558103 24	1.688591273339 38	1.668979758379 42	1.663882306505 5	1.667702140151602
IRPC	1.302994413638 24	1.312143083297 5101	1.291213906648 52	1.308213613562 78	1.294818704794 21	1.315942760226770 1	1.303535023144 09	1.314230802826601	1.291199595572 16
IVL	2.169674158224 62	2.176127604656 4597	2.099208481519 3	2.163265147476 96	2.121105430045 0503	2.149670065325 02	2.192152291535 88	2.175706943140 2	2.098521068531 79
KBANK	1.390784419662 55	1.388481022626 66	1.380176111355 55	1.383164651846 001	1.393341435389 599	1.393340704343 03	1.387687771936 901	1.388041575605 87	1.380321850507 95
KCE	1.520371921829 45	1.503464990523 73	1.481392745590 2	1.495496427804 27	1.487726951920 28	1.481796531370 31	1.536164453294 6	1.515045513143 75	1.481028322445 94
KTB	1.182350966393 83	1.175296928105 07	1.162353069534 7601	1.217181421068 02	1.183908738625 85	1.194715891080 51	1.183274838188 12	1.171299844462 86	1.162218545101 29
LH	1.246904401497190 2	1.238772009544 53	1.217443052413 46	1.218880058036 75	1.216235758907 55	1.197404535779 54	1.256652830695 57	1.235383120421 23	1.217567710975180 1
MINT	1.653853941466 91	1.625797053651 65	1.624206600109 98	1.646586190280 91	1.610779761616 126	1.622263493033 09	1.652938947906 75	1.626384691398 1	1.624031982147399
MTLS	0.942870883314964	0.949695556065690	0.985621253110471	0.910922166922948	0.999676307463425 1	1.068656007350 86	0.943929066568641	0.950377188950938 9	0.986202445955184
PS	1.337717858128 24	1.293903110731 83	1.283913000085 42	1.375653041032 68	1.298169540744 3	1.359636641226 93	1.329229682211 2	1.294706082066 81	1.284192400094 42
PTT	1.940009352204170 1	1.910020794871 85	1.918503990621 83	1.899822720843490 1	1.868680271575903	1.847580831529 98	1.941866174707 95	1.908252689999 55	1.918521313278069
PTTEP	2.496455754126 39	2.461802676070 96	2.343288575479 99	2.281678199334890 2	2.240123769971 56	2.194586775189 4	2.486498536490220 3	2.445673861271 05	2.343414372783 66
PTTGC	1.803481632286 36	1.796062974309 26	1.806391156674 66	1.852675676291 32	1.844288672392880 2	1.836722100586 3101	1.792469303258 27	1.796877116828 3798	1.806514664704 06
ROBINS	1.503883341272 09	1.514572109178 52	1.493445739778 06	1.495814800368 5	1.486847047547058	1.485389241430 83	1.505259109737 37	1.515529919265 14	1.493510894956 64
SAWAD	1.519465282402 22	1.427643938646	1.413948587150 74	1.509102655400 09	1.445325321114 28	1.447335925754 28	1.510661760474 85	1.415935775827 869	1.413197778730 787
SCB	1.480922675400 4	1.455450630728 37	1.430356472585 99	1.447688770673 27	1.439557809166 66	1.428825895301 57	1.474232016252 55	1.460569262240 4198	1.429917331057 25
SCC	1.090361339983 95	1.085460801545 91	1.090605238670 87	1.109252009004 39	1.101931924601 76	1.113603127670 9	1.090743846876920 2	1.085705841335 7301	1.090591424099 56
TASCO	2.356121560460 64	2.369366075187 06	2.322180863776 35	2.434626746626 08	2.438975993555 27	2.441089517686 3	2.348017853888 67	2.366954153949 86	2.322115367827130 2
TCAP	1.136137297293 25	1.125288465521 51	1.123731370131898 8	1.162843780120719 9	1.147045474326 78	1.147579768607 37	1.133617968066 12	1.124969831924 31	1.124010032211980 1
TMB	1.443276788899 118	1.434682701071 15	1.441319721655 82	1.428084389145 9	1.431361115696 69	1.440801543350 67	1.449539596134	1.433866473464 61	1.441395240677 651
TOP	1.497817242497 17	1.501921033853 56	1.481976677097 1698	1.502794992591 09	1.500792032443 0701	1.489240684340 8	1.497952601894 9	1.505346667121 95	1.482017655832 54
TPIPL	1.672048500687 26	1.703966697559 97	1.622049915196 0999	1.816179050526 27	1.795249389182 9	1.758876717555 7198	1.676873549606 2601	1.687602026250 95	1.621099824300 22
TRUE	2.042013960924 61	2.025299221101 83	2.054463178276 4	2.209223451854 8	2.113710459411 24	2.174916056229 7103	2.042976534540 59	2.028205510585 92	2.054314798390 88
TTW	0.853094224194484 1	0.844509884878823 1	0.841127890726594 1	0.855820410011352	0.853717483545326 1	0.858786695247571 9	0.853035123419415	0.842969997093302	0.841568247999842
TU	1.375690763149999	1.351067654669781	1.354479123300877	1.379108133080 92	1.331203905934 12	1.332903256846 3	1.383886386878 05	1.353257522824 1	1.315508033305 24
WHA	1.607059841773 31	1.569208158990 95	1.587938512086 82	1.691211455634460 2	1.645524283350199 7	1.622144365489229 8	1.595686874983490 3	1.568313530709	1.587826307752 19

Table 18 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.76387708826563	1.75451068049274	1.7470265735811799	1.7695649133226299	1.7622155584489998	1.76246518859179	1.75188991194974	1.747077047888	1.747079941443097
AOT	1.0488532209711399	1.03844052251942	1.052792748447795	1.074125506478897	1.06243343504532	1.039894022986643	1.055125024233157	1.037908738324078	1.052604311981 01
BA	0.9284941910297331	0.9019988393202301	0.914259185264729	0.948543664274789	0.968211652182157	0.971725076623108	0.919328661478387	0.9048837459832021	0.915343636871732
BANPU	2.073206900085139	2.053487788345852	2.014219887749032	2.040944771450 09	1.9940346578292598	1.9866869666398799	2.080508569227 88	2.0497625567298403	2.0143137178796797
BBL	1.2099251903994201	1.215434771192 79	1.22547537988498	1.212076302617 65	1.22176351687572	1.22842709199291	1.213611831584 76	1.2152397660197902	1.225384633117 17
BCP	1.370552892758 55	1.351387814172 79	1.350955217737 91	1.417562538207 5299	1.384310167074 77	1.402530858140 9202	1.370478827837 92	1.351193864362 85	1.351041673668 83
BDMS	0.9282714321547959	0.9139636438950121	0.9069453701390441	0.9354920028096829	0.9285056113167 92	0.923047334958 91	0.908987636299377 1	0.9105740394636 62	0.907029677727704
BEC	1.7599049167513199	1.749659291848 18	1.703172983565 8	1.788525036927 4401	1.759371473833 817	1.759080330270 1699	1.767350088515 28	1.759373941273 0802	1.703199080809 625
BH	1.468114556005 56	1.438912403311 42	1.453926042460 3602	1.519566279 735012	1.504134476 4614801	1.535920509894 2599	1.458037089285 52	1.435570470171 1098	1.453432187916 77
BLA	1.6471292673036	1.618073195894 9202	1.616237465666 22	1.762660183264 4602	1.753333614123 91	1.764356315033 9	1.664910355614 17	1.618066051923 8201	1.616362060679 692
BTS	0.959523460257505	0.941124508341628	0.941199519114301	0.9569413341556241	0.928426326848 5	0.943718082210878	0.952727067061 4469	0.940405736146 789	0.939901730317 744
CBG	2.1192123752344303	2.049594331097 78	2.024195780969 6998	2.172800917017 65	2.108682207161 44	2.184740211002 7	2.084331594698 73	2.054545766165 15	2.025008411460 5897
CENTEL	1.514685332658 33	1.507721094066 599	1.525568894178 01	1.555759981868 61	1.541223507060 1	1.564192339519 2099	1.523310076238 74	1.503962319669 03	1.525506994677 25
CK	1.367350328136 74	1.357858980388 21	1.347053558262 01	1.438234022809 5	1.394294116600 32	1.371839168804 98	1.371816924347 0701	1.361249350764 92	1.346202034603 22
CPALL	1.401850008433 781	1.362453443677 1398	1.351571837979 49	1.393511771405 96	1.360416676726 402	1.359184967117 21	1.395472739739 1902	1.357017388595 58	1.351567761521 6402
CPF	2.065186100417 53	2.043415084537 9	2.013700224947 61	2.109581327616 51	2.050452849274 63	2.062135085920 42	2.089288591262 2	2.040184060263 09	2.013738857687 88
CPN	1.291911685740 57	1.278361974185 32	1.265204255268 94	1.253472549635 55	1.239969811994 77	1.251906552446 78	1.293291450198 3	1.277421358458 62	1.265290118464 24
DELTA	1.855107316414 91	1.806440702996 6998	1.784546344487 6202	1.786200217072 14	1.760229200180 7602	1.760318702982 4299	1.829414477271 0599	1.798064811016 5	1.784513052212 7401
DTAC	3.1265247387802098	3.134731534297 7704	3.088828686076 63	3.136221357653 8602	3.134597590417 01	3.164340895800 74	3.137834258250 097	3.129214850569 21	3.087987642849 31
EGCO	0.921083293127376	0.915754525028967	0.913383146427831	0.928425588528 1509	0.915066209602 6579	0.927291483935 3399	0.924392294722 1509	0.912875670187 7411	0.913447796082 465
GLOW	1.360030767503 4702	1.330595373163 8	1.276838154138 64	1.309176033142 35	1.318904767797 68	1.313315221301 27	1.365892453632 1	1.326541867523 19	1.277120375250 6601
GPSC	1.482503335287 13	1.488051299906 86	1.466209945735 146	1.732268489371 902	1.790437057906 01	1.705508626387 01	1.586858764427 96	1.496530420137 47	1.680158171831 4
HMPRO	1.495548235460 84	1.505767805769 42	1.495923524866 67	1.504610322435 5	1.491495881676 91	1.477054731828 1801	1.495465438962 54	1.503279132110 13	1.495757350472 301
INTUCH	1.66730632480671	1.664744762686 71	1.668120679840 699	1.731625051705 07	1.696434423824 46	1.688591273393 8	1.667931782356 78	1.663459116258 15	1.667702140411 602
IRPC	1.292486733603 53	1.301336284843 75	1.282318703466 93	1.280534838567 71	1.289509345236 98	1.295599783115 37	1.295260624720 64	1.302361122944 44	1.282593873973 701
IVL	2.1499376717281	2.107965136213 55	2.064700855887 96	2.157050056497 74	2.13762361670 16	2.173733156737 33	2.127447863966 803	2.114514787673 06	2.063855911762 105
KBANK	1.3956279878659101	1.391812926903 7	1.390679213015 9	1.415608663063 81	1.401231914648 09	1.417837783837 9	1.396068400524 4	1.392420319070 47	1.390860654624 55
KCE	1.517746789079 15	1.509908105448 201	1.477917952599 7599	1.497592949180 05	1.489399519717 64	1.479004570474 04	1.536362343127 8701	1.500998441849 61	1.477664334910 12
KTB	1.182385971267 61	1.179626886605 464	1.162328853480 98	1.209222747183 49	1.185880998380 2499	1.194715890185 55	1.181876230597 9	1.174517713292 11	1.162201067225 32
LH	1.251328441802 03	1.241409311202 8002	1.226806383173 82	1.235196171554 07	1.230888268018 81	1.222079173752 04	1.256223862669 75	1.238968083177 77	1.227095151473 59
MINT	1.6961717409259902	1.663340929446 99	1.633495908882 92	1.698411198955 1303	1.658911676267 619	1.631638527000 6302	1.702162894149 19	1.663133292700 0302	1.632678494373 6399
MTLS	0.9413674884059 22	0.9501584565210219	0.9856212513109 46	1.013103477539 8	1.005224630617 7	1.006656007350 85	0.951270960563 919	0.949662987233 65	0.986107068966 011
PS	1.383903178928 97	1.341970238272 9	1.329655815263 29	1.318285833670 19	1.288100571058 22	1.326577327902 43	1.384859722344 52	1.341734421269 33	1.328874688333 03
PTT	1.942391427229 46	1.899172796126 02	1.918647982483 3	1.902566057477 6	1.850749922757 9798	1.848373536826 3363	1.942954746847 9198	1.907287718220 5	1.918666711403 68
PTTEP	2.440484419571 13	2.335246422121 73	2.293526352645 42	2.261304238840 14	2.221940465990 5297	2.212849413011 94	2.411285950379 64	2.336191287119 3597	2.293948978572 81
PTTGC	1.790286822846 25	1.791897462982 3399	1.806391156674 66	1.852921650724 4002	1.839613836990 53	1.836722100586 3101	1.801367130961 48	1.796883279120 92	1.806514664704 06
ROBINS	1.5027205701044801	1.519952338345 6002	1.493445739778 06	1.498226208626 92	1.487106964592 598	1.485389241430 83	1.499934759185 1	1.517323515290 99	1.493510894956 64
SAWAD	1.4963818051565099	1.423458822587 345	1.413948587150 74	1.514234678743 34	1.446398529599	1.447335927544 28	1.498955587125 46	1.424298293712 86	1.413245522191 25
SCB	1.490675296114 46	1.461327727717 1999	1.430356742589 9	1.445972564383 78	1.439715997455 49	1.428825895301 57	1.472548907873	1.456787216004 93	1.429917331057 25
SCC	1.093148607519 6202	1.084052434566 022	1.093099495510 84	1.124913369021 59	1.117782244717 63	1.133227342190 29	1.091734960937 07	1.084099839143 59	1.093093843053 24
TASCO	2.39959124279004	2.384017058524 92	2.355427022705 403	2.686682235900 265	2.637302765256 59	2.604837708099 41	2.396598956565 6898	2.379345513729 26	2.355694585905 599
TCAP	1.132020515558 87	1.124709463550 08	1.123731370131 8898	1.162748156358 85	1.144035008909 19	1.147579768607 37	1.139231687802 9	1.124917311634 86	1.124010032211 9801
TMB	1.436309361927 12	1.436161761598 52	1.440279663828 699	1.439455884543	1.435686803907 5	1.438825804428 7698	1.433586190361 93	1.434077549192 3201	1.440329952196
TOP	1.509006183126 81	1.503800883928 66	1.481976677097 1698	1.502851034956 33	1.499256141609 83	1.489240634308	1.502723095905 94	1.505491933078 76	1.482017658325 44
TPIPL	1.719782709188 39	1.691585799025 08	1.619156006328 93	1.924313628970 36	1.881540657515 65	1.844695250931 45	1.723470419924 1901	1.712405686245 8599	1.619445159551 601
TRUE	2.013484081835 85	2.001899830168 2	2.025129902372 2003	2.024615765115 1	1.993516752955 0097	2.041137050392 17	2.043067750508 4	2.002628834842 1803	2.025130439554 7
TTW	0.867253304884622	0.850964185591899	0.843686943948312	0.896428533486599	0.872942533846599	0.877991939495 3381	0.863654575043 167	0.849926865846 45	0.843691264002 859
TU	1.384812395826 16	1.352236500575 31	1.315479123300 64	1.379466503290 36	1.335730015715 2001	1.332903256846 3	1.385697445130 2	1.349570724887 59	1.315534122402 34
WHA	1.592424565874 27	1.567007695946 702	1.586006836616 14	1.688372443465 902	1.649582326186 71	1.622683195222 5	1.611755492251 4699	1.565889450982 501	1.586016033868 61

PredictPrice7 : CMF (Chaikin Money Flow)

Table 19 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	32.6680777417099	32.8275390488785	29.161897324964198	32.9165403414302	31.9687556432610 98	30.4689548625927	33.3000380676 87	32.5392443550451	29.302540989940002
AOT	29.001673763768	27.926061267855 9	16.67979424013 87	28.093591613911 5	25.465607740039 296	17.07027084164 17	29.158408601345 7	25.923262102470 6	16.576933588335 3
BA	2.1777504152284	1.856277492969 54	1.473858389221 9599	2.109127594631 97	1.874535024016 17	1.170911643974 38	2.164362720554 84	1.998392176852 5699	1.530193564676 79
BANPU	42.719418796473	39.065947707999 6	18.485659978801	41.700669494199 8	36.698056823089 6	20.663349067531	41.973437798522	37.9640095503 4	18.618789620105 8
BBL	7.133212934123 09	7.015777290952 1	3.294330641410 12	6.340906970867 95	5.865713649009 62	3.753923610489 2897	7.435846176255 009	7.012305760337 12	3.275622666768 155
BCP	1.744534033644 5798	1.732761653291 2302	1.496489654664 89	1.942281257970 6201	2.087059526890 86	1.479144685591 1499	1.791772266755 1699	1.764827998429 6	1.497512401697 81
BDMS	17.03374727515	16.696903369334 6	6.732201086185 12	18.553058153806	15.560445416390 401	7.003977788645 891	17.998570725989 598	17.376746158173 9	6.694565275206 2905
BEC	35.192535404664 2	33.872972215584 504	16.539945323407 8	35.607921596259 5	36.773102148314 1	17.337043139251 097	35.390818606200 305	36.532605536051 2	16.575442892236 1
BH	3.1381126933375 1	5.3189206376896	2.3867055186405	2.578660738893 85	2.2919578541505	2.536751137539 85	3.152255905542 85	3.1264003943647	2.383734775711 291
BLA	18.259023780000 2	18.012998789807 3	7.856855539028 97	18.887765541884 598	16.254655928064	9.822262244690 49	18.098418320517 9	19.683818686502 9	7.738028338077 12
BTS	1.417268502688 55	1.411860054986 66	1.011493959771 21	1.322893887478 07	1.549405881972 27	1.021331211563 36	1.413153534010 8499	1.575581817419 0899	1.009291041952 32
CBG	28.415760170000 098	28.164731438871 797	26.073767195562 198	28.458861957792 298	28.134291640288 602	25.903700529548 1	28.414325920541 3	28.306385573352 2	26.018844206519 7
CENTEL	8.708796671593 18	8.180093447308 471	3.70443131570451	8.247561229373 99	7.832837618882 99	4.412966628108 74	8.411622976141 4	7.125197982413 4	3.679041494902 46
CK	5.397230122705 63	5.697025526750 73	2.921073787225 9798	5.581692841138 95	5.918094460155 06	2.988388652429 78	5.250075872100 15	5.517296518021 55	2.865300673205 1
CPALL	2.893463329620 33	2.707908824101 53	1.979445590797 4702	3.175059335111 34	2.800243463279 78	2.131475076466 43	2.825973211498 09	2.528298591681 57	1.980066209354 39
CPF	8.602368511348 36	7.602998149442 52	3.977501179402 4295	7.904556928957 571	8.177120634006 9	3.967844406511 1402	9.057636811987 52	7.962600970007 1	4.024208515991 16
CPN	10.581390582121 2	10.498805569110 1	7.838744225711 309	10.827354448050 5	10.610871051885 4	8.868285573299 24	10.575303323003 78	10.257061407000 501	7.765040457921 9305
DELTA	1.842863370616 81	2.035510647349 77	1.939265748382 599	1.844817195956 02	2.100452526083 898	1.960883874494 2	1.856708074109 14	2.023826583029 59	1.937106074346 41
DTAC	144.04338987088 8	146.92966818046 9	67.1928343664053	144.43120805224 8	126.17560311424 6	70.01955519683 591	145.33207862113	137.25461436113 5	67.250139943727
EGCO	9.236409909756139	9.173542685199 58	4.252974988933 283	8.466825667798 659	8.684507903383 16	4.418686312890 28	9.313892893911	8.260040288667 21	4.247719596089 29
GLOW	1.639255307782479 9	1.926152232705850 2	1.484339553115377 02	1.734812817814 05	2.040719295472 17	1.585630753345 06	1.680236238924 302	1.901133309856 71	1.484657388538 55
GPSC	5.855180793856110 5	4.993384597374 62	4.936921942525109	5.61086318030 69	5.044716830647 2205	3.797494761109 36	5.867932755764 9	6.044603446350 43	4.974581091801 39
HMPRO	6.819035867454 35	6.680248119540 19	3.836423434662 3	7.393136884908 62	7.095208949571 95	4.275714533375 7	7.018911901912 995	7.688080542108 2	3.833145094242 0298
INTUCH	29.639248181753803	28.785413330357102	27.085078920297	29.669372651022503	29.150994418911	28.539505567484 902	29.496831129336 798	29.417759362045 796	27.124058689041 703
IRPC	11.845799942500 4	12.537204773208 1	5.775941667855 959	12.167238648411 601	11.774376667954	5.595333400089 33	12.322940508462 4	11.182059528014 0	5.802998191434 78
IVL	8.349303519158 63	7.610034711495 195	5.192333941985 17	8.59136971401 06	8.757143391105 88	5.626948790127 04	8.053246050211 93	7.906259218660 45	5.184564119253 17
KBANK	2.792235748033 098	2.869752224752 96	2.289912612641 65	2.997528205071 4602	3.197000093433 241	2.590111415571 84	2.923602226929 07	3.496862120391 67	2.298561317373 994
KCE	38.750945073994 394	38.216065886782 3	24.933036328069 4	38.263562663358 2	38.851479193027 2	25.145081012576	39.196933209494 2	38.382847634332 7	24.924216177175 2
KTB	1.611460001454570 1	2.146611973675 19	1.245896909631 02	1.546292392216 601	1.813924594355 4	1.296388418674 33	1.535959756895 44	1.773842607458 13	1.250317810365 75
LH	1.384494434422 01	1.453958228894 65	1.390072292328 17	1.341287179273 16	1.425883916026 8801	1.355258673285 92	1.403535201145 3899	1.449364818244 6899	1.390154305469 0202
MINT	18.031409725160 5	17.705600734175 9	9.177374280623 42	18.869183622579 698	16.664515432962 3	10.169836222550 401	18.297835915971	19.528512901355 597	9.095581657757 4
MTLS	3.868328175829 6796	3.376706314914 71	2.708452638265 77	3.512918096670 1	3.659022512256 2403	2.475894074258 68	3.787882189778 39	3.489873906336 95	2.700430926731 44
PS	1.374976608258 52	1.544588937449 7	1.354435848993 33	1.378941258061 31	1.576865357246 61	1.378849695755 4	1.324228135653 202	1.563754741927 602	1.359903007843 46
PTT	7.689149865272 101	7.142778405273 18	4.438621598090 52	7.790035163361 8	8.161851401113 24	4.748201284971 261	8.294485323205 29	7.703030131746 35	4.437807477626 52
PTTEP	30.782117372450 003	40.594995127019 104	11.281680485987 5	29.765208769349 3	29.813595936356 2	11.235736362633	29.797936746545 2	32.037121077379 2	11.339012149474 1
PTTGC	4.209077778381 12	4.357308195551 61	2.910308967776 71	4.763291129144 76	4.560773162578 86	3.207495188296 03	4.759922043551 771	4.582782972494 1305	2.906666472590 53
ROBINS	7.105876506145 6	6.578263212397 981	3.946894820529 5904	7.126503200038 97	6.443174573563 58	4.239530006869 86	7.522296032955 7305	6.802359925815 191	3.957391468739 7598
SAWAD	2.701625145421 58	2.736250696786 63	2.050209744874 89	2.749260691387 74	2.579600169534 18	1.966392376398 57	2.603497728257 39	2.856480018247 09	2.047783064666 5502
SCB	11.696499199464 599	11.465613883551 4	5.467309715869 92	11.351942364412 599	11.764625831692 1	5.758452762444 6	10.971875781024	11.993186136383 4	5.466023876843 79
SCC	1.161993669859 101	1.241097144682 09	1.131423718672 82	1.166052527515 97	1.273104371071 87	1.165138135077 38	1.160630093086 7	1.215421310580 77	1.130161629062 16
TASCO	17.029488246667	16.007652372710 8	9.941369339698 68	15.855922338040 8	17.525294668407 902	5.699375049185 705	18.760831161582 8	18.805695465591 97	5.988906746721 88
TCAP	3.544839024645730 4	2.859097859312 38	1.854688251740 4	3.477573454716 7203	3.176366366690 1603	2.143692563295 8	3.615988347845 503	3.162932209000 243	1.854855526142 202
TMB	1.496906301902 52	1.582803485035 73	1.503085315829 76	1.538445692904 4898	1.525427426544 3502	1.559736038405 94	1.493723212640 71	1.635621190629 98	1.502767696869 29
TOP	20.152299140113	20.384996187910 4	13.180388595016 302	20.422450486356 2	20.019918835012 803	14.196692669195 901	20.125736382381 202	20.636116356957 5	12.951062898380 2
TPIPL	2.540034190575 7102	2.608380824908 2097	2.183588323178 098	2.498701053000 93	2.715466612345 15	2.124191455385 9798	2.350041745639 87	2.723223120092 63	2.186447778848 081
TRUE	4.522774591863 37	5.817611840238 18	3.268367801506 995	4.636657687614 02	4.952035393390 08	3.198072169644 12	3.649018419840 6	5.115804534200 3	3.266128221609 297
TTW	1.969054342183090 2	1.987582474456 61	1.623737409631 17	2.209120298671 75	2.012255567653 6797	1.691510785373 8403	1.991596950020 86	1.976971289260 19	1.656073554394 96
TU	2.051046499093 12	2.143878338151 13	1.641860766697 08	1.865576897564 9598	1.898297643331 38	1.538426251336 4	2.079671726558 6967	1.969300074231 66	1.642456441029 3803
WHA	1.905986533944 82	1.926259619705 66	1.601446107029 4002	1.945201505968 8001	1.932334617535 84	1.705907556949 4	2.027537441856 9	1.953998456889 99	1.598356559964 157

Table 20 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.797645551625525	1.8220673260868299	1.760150929120304	1.76815105319435	1.776625147217 19	1.7694435066749898	1.811535159431264	1.8248790377534099	1.7598319720125901
AOT	1.04978272368437	1.04168223079775	1.03072504786548	1.03158171414521	1.0308671177478	1.039018173465 87	1.04336803323454	1.0389895103255 8	1.0300611463991 9
BA	1.0847266239067601	1.0808904053177602	0.992792378806547	1.07909225444285	1.0948377888310599	1.175291685847 28	1.08238896723729	1.0732058964047 1	0.9931962536332011
BANPU	2.0502236733036	2.0716797086764003	2.0143079490355897	2.03537300498036	2.0446591364584	1.9773243675569998	2.0582006350125 1	2.0814962923717 1	2.0138813709377
BBL	1.25300283864176	1.26415368409028	1.22057725981649	1.2406692315791799	1.2527701852448498	1.230471856474 37	1.25005358841 79	1.2564891691657	1.22003957180304
BCP	1.141616814913908	1.4254750725176801	1.36319429454319	1.49559676725352	1.5055554920068 6	1.315926710011 3499	1.42072062678929	1.4179452139411	1.361439359167 82
BDMS	0.9485562793338009	0.9243623644244977	0.887834289115991	0.960326843549025	0.9386190826160 9	0.893643055634227	0.935848064232643	0.9308617805816 38	0.88444843448465
BEC	1.7129067679558	1.7736052333214398	1.65761832219033	1.7845735079241702	1.7766839452108303	1.676427835915 47	1.7355627860423	1.7674981774728302	1.65763998302024
BH	1.482075855318938	1.491680372427 83	1.40934963958513	1.474815868893 96	1.5195797234995 2	1.481214441484306	1.474004966237 65	1.4950905791403 6	1.40927686990165
BLA	1.732445713933334	1.71930000597524	1.6400990195971799	1.795762547444 76	1.7880238783284 7	1.718516169569999 02	1.732784518619 2898	1.7189141752719101	1.6397904687997
BTS	0.938146642819361	0.942008769779827	0.925896654900311	0.964521310642247	0.9674146738078 06	0.968020226484367	0.933338056012294	0.9298536209098 02	0.9255448040787401
CBG	1.969058017126 52	1.995323037151 52	1.9370688204751	2.0642191636066403	2.0529897961301 5	2.039821021964 9	1.989732789131 37	2.0203555001945 19	1.9367765504650198
CENTEL	1.568054324178 6698	1.586517915495 32	1.506062276188 36	1.556836557993 48	1.559014730759 47	1.539049566080 15	1.547168359060 384	1.585334090034 55	1.505681829073 76
CK	1.355891590704 78	1.378167264821 19	1.335233298421 21	1.40623602570491	1.3956629189595 301	1.332113966281 3402	1.365882237356 3502	1.3805095620932 7	1.335822104635
CPALL	1.31647755096145	1.355108715326 08	1.321164459214 03	1.32704773088773	1.3535644926844 999	1.341478811806 371	1.307193724552 05	1.3404159223909 6	1.32100343311 138
CPF	0.026523267593 19	0.0427059811997	0.0153958286524	0.027414066039586	1.98293972529602	2.068711724733 54	0.030556358118 5098	0.0393290317855 6	0.01465424325704
CPN	1.272956824487 78	1.295188154680 0201	1.238777247250 17	1.2076403632361 798	1.251657049828 5	1.207036542131 93	1.273665085783 44	1.3062928566131 1	1.23873641641073
DELTA	1.791766663220 98	1.809998721759 84	1.711482346946 03	1.793756282998 09	1.769887101177 62	1.699728508569 46	1.797240712362 92	1.806607002876 32	1.71457235015908
DTAC	3.400751067504 12	3.401263032852 5103	3.311569856008 282	3.582731075057 06	3.516594129026 797	3.275495109028 11	3.401104318171 3303	3.385916196490 7902	3.31514451660 159
EGCO	0.933736225179412	0.9585883751768 88	0.914976419342353	0.9888741869198491	0.9832853833801829	0.914318633373171	0.932643838741604	0.954506117669288	0.91481761452910 99
GLOW	1.3007354722856	1.323485111201 35	1.243020335541 46	1.30546866524798	1.312589304224 96	1.263016503 93988	1.289280564787 35	1.356533737327 59	1.243095658917 69
GPSC	1.401103232677049 8	1.437764355740 653	1.375447629626 1702	1.738522326493 1902	1.709740058355	1.620647376274 53	1.473176209358 16	1.447523048547 71	1.39064290107489
HMPRO	1.565333003252249 9	1.591266385603 78	1.489144790871 73	1.514801683280 7598	1.512457775308 41	1.477177969665 66	1.565760182032 17	1.579690064410 35	1.48894438427422
INTUCH	1.625397968537780 2	1.651783838581 47	1.602663609503 49	1.644668714548 09	1.692024794102 06	1.605642450961 3298	1.628676694714	1.657362454499 294	1.60187589253498 99
IRPC	1.328923235860001	1.365193842120 4501	1.282903112961 8	1.349397771563 19	1.330312875837 86	1.308591503043 2	1.332031461034 68	1.351606034986 35	1.28279849791672
IVL	2.205508193219 88	2.227459197129 71	2.091543638951 9497	2.245690275041 94	2.224481744048 7603	2.147935117873 83	2.207537458439 04	2.224758577660 8	2.09156278305635
KBANK	1.39136216264938	1.416643043405 35	1.402483672950 13	1.477167426080 2	1.474094396890 32	1.460161945835 04	1.388426543389 18	1.412605748043 49	1.40245095806423 01
KCE	1.492216143013250 1	1.503526816570 22	1.465855228328 33	1.492389309364 53	1.464596252852 09	1.437442166925 53	1.493792437076 79	1.491427536676 83	1.46595522435630 99
KTB	1.198914565208 59	1.228867565533 901	1.151938909214 74	1.199649914390 81	1.198271684843 8101	1.184561912507 3699	1.199604255066 77	1.222815080231 2499	1.151365552419
LH	1.252741772662 2299	1.233467634082 68	1.197421883632 1	1.226471810388 39	1.276623532987 77	1.202470771269 6698	1.251768620486 0898	1.237338529364 53	1.19738592713481
MINT	1.593333399024 49	1.628380676793 36	1.593230325260 63	1.585141254749 04	1.598302976582 06	1.598035183759 7	1.667566090069 23	1.626694920071 73	1.59330065180777
MTLS	1.018196842022 2298	1.042709415412 29	0.941435117441 748	0.985479871035 5069	0.986855020523 095	0.955108669334 203	1.024523060867 28	1.041884694132 04	0.94255806923366 9
PS	1.233413266725 8701	1.284222557929 89	1.206284722665 91	1.240031599397 34	1.259006473741 56	1.201047361708 95	1.226464887445 66	1.270890649169 35	1.20615898013458
PTT	1.922185315540 86	1.953821154620 1198	1.876870431639 62	1.826882938546 66	1.824764921859 1198	1.812532497871 9701	1.924144806786 0298	1.943692275049 3701	1.87896936275516
PTTEP	2.367508581842 5	2.375063330370 4203	2.258631750360 2403	2.2056319669836 23	2.220532465199 16	2.184936084623 37	2.380103114350 8102	2.347762734167 94	2.25889625787 52
PTTGC	1.849902443854 05	1.873696025711 56	1.776484264902 5998	1.790484334829 84	1.779351504434 2698	1.820656138471 3198	1.843798389485 2301	1.869297252407 15	1.77670926820752
ROBINS	1.488572428247 89	1.529231188851 553	1.447422295592 21	1.460047386087 79	1.497468280187 3	1.441870745758 63	1.486416003582 91	1.531611133825 81	1.44732079760801
SAWAD	1.520138048255 74	1.523422208278 53	1.408643773575 81	1.469922210492 1499	1.460415287471 8	1.390445687193 49	1.518235380146 32	1.517545680902 53	1.40867048354729 9
SCB	1.469852177492 5701	1.508642989701 93	1.456869499283 92	1.477733376533 62	1.484327662923 84	1.467831369188 78	1.467652132051 63	1.510260465551 37	1.45690678159972
SCC	1.086352836224 599	1.089702439336 87	1.069141081376 3	1.089153484898 81	1.090048049375 97	1.089607406380 52	1.087923712701 62	1.100912831448 0598	1.06935954420872
TASCO	2.372407877884 81	2.396795984787 75	2.338346613022 81	2.510506808429 8203	2.495283256210 93	2.439168864451 69	2.371975477678 76	2.397163060999 9397	2.33711911202266 98
TCAP	1.201807632075 25	1.216011729613 52	1.129449522604 86	1.165499265330 4	1.164850230258 96	1.127122366916 67	1.197174538904 61	1.209305229073 01	1.12952371411 395
TMB	1.508428690268 72	1.527309610050 3	1.392271760763 4199	1.505790892927 2499	1.534767690582 85	1.510857826809 1	1.490998973873 07	1.534602556676 27	1.39170844036 15
TOP	1.549378883275 78	1.566033935100 8998	1.517537453815 1498	1.597284307006 2	1.598597090317 0201	1.514003950414 3	1.548331227039 26	1.565897808406 07	1.51755415772020 2
TPIPL	1.669762883664 69	1.670002267917 22	1.587917632313 0199	1.876917112104 1601	1.858737598852 05	1.662846391001 11	1.590357823650 72	1.694586306656 2699	1.58750328282352
TRUE	1.998134116861 66	2.082112666077 51	1.969261738890 92	2.138195776751 04	2.153169073964 27	2.061684013312 06	2.033803047234 11	2.054036724804 89	1.96973756566 054
TTW	0.861315895743309	0.888279991702 1021	0.817195437810 1689	0.833685249903 881	0.828423333729 871	0.839038752960 03	0.859528562838 5361	0.882414406264 299	0.82136187541802 3
TU	1.393076674820 298	1.409824885673 36	1.364088671035 75	1.411316658112 85	1.411470316522 75	1.394829840457 26	1.400655109263 33	1.408380412200 37	1.36421529896 206
WHA	1.565521670900 9399	1.629780495559 17	1.519809843746 7	1.605740560058 5	1.639462320950 9	1.604450481820 097	1.560228972133 01	1.612246405641 5	1.51987023218825

Table 21 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.762562856546572	1.764104780313 78	1.778935073930997	1.7335135897315002	1.7377815764510598	1.78636260086057	1.75987759948539 01	1.7638377650616301	1.77716498109945
AOT	1.073958420473290 2	1.060364234982 67	1.022551948244 78	1.047126997671 1999	1.05197297386873	1.039139118374 31	1.069956527022 3799	1.078678011242 91	1.02261323336258
BA	1.064971865968 62	1.040778462630 26	0.943788777945104	1.023184689304 04	1.011406775489	0.951712687797 069	1.040563157202 09	1.042443841406 58	0.952498858150242
BANPU	2.021818959587 971	2.059916621089 6502	2.007475930234 33	2.030421033182 3302	2.040638831156 24	1.993259336744 598	2.029813263856 34	2.059480330047 66	2.00758855464703 02
BBL	1.268788934891 0001	1.274851835284 402	1.217302581981 23	1.240042945906 02	1.244916195555 21	1.223764777156 09	1.268487023118 3	1.281447021782 3999	1.21706725669979
BCP	1.425561225863 93	1.419101353024 06	1.363194294543 18	1.514077173982 1601	1.504847257124 22	1.315926710011 3499	1.414812228865 78	1.423589078615 58	1.36143935916782
BDMS	0.937719999714749	0.925241350339454 1	0.887834289115991 9	0.960446560394899	0.939760570105 3	0.893642180605 7989	0.937039640848 1099	0.925282927010243 1	0.88444843448464 89
BEC	1.711747171921 87	1.728682842865 2702	1.660441123665 15	1.794311205297 6402	1.779049620982 95	1.673476779364 6101	1.714072164840 7902	1.728552973963 0898	1.66047443020443
BH	1.451404152662 09	1.515288807185 15	1.420317836191 02	1.484276720777 302	1.502683232209 58	1.490280404893 29	1.455568638203 98	1.523997022995 51	1.41822125382586
BLA	1.731835670971 5	1.719717875910 5698	1.640036175007 28	1.814246886355 73	1.790986184512 6398	1.718516169569 97	1.733317267161 7498	1.719367174366 7398	1.6397904687997
BTS	0.932320082348021	0.936621672870 499	0.917207044357036	0.944952082791681	0.933084811851 7411	0.969429020513 965	0.932201417923 83071	0.936462573562 361	0.916949653828102
CBG	1.997553984650 87	2.074617585065 39	1.937068820475 1	2.085639906868 9804	2.063043964897 303	2.039821021964 9	1.983943161301 0001	2.014558827524 25	1.93677655045019 8
CENTEL	1.571820529684 499	1.593986908462 101	1.501840983854 53 01	1.539465939315 65	1.563771410806 8999	1.545080197384 36	1.572692087731 51	1.594369138364 64	1.50188675951137
CK	1.420627975665 61	1.424596744966 49	1.355995174105 198	1.476859817198	1.514766918724 9	1.360616233436 5899	1.417329487498 701	1.464440930636 33	1.35405323491317
CPALL	1.317807316781 6	1.356677009352 1201	1.321164459214 0401	1.329817530659 17	1.354107090782 58	1.341478811806 371	1.315670695043 39	1.354417179930 36	1.32100343311 138
CPF	1.980931118384 26	2.021981309569 5	2.020697426514 9302	2.019267026564 6	2.009750040256 9	2.086566264840 76	1.981808679584 42	2.019890181445 03	2.02042169500334
CPN	1.277797410584 5902	1.294486719084 43	1.238777247250 17	1.207184812301 22	1.242776272320 95	1.207036542131 93	1.273100605170 1899	1.300989786000 51	1.23873641641073
DELTA	1.785084171397 66	1.811105236374 03	1.714482346946 03	1.790455658520 3	1.784638947987 2702	1.696728508569 46	1.787856650629 2	1.814619996843 31	1.71457235015908
DTAC	3.394367834948 56	3.343665179860 3703	3.310431797766 02	3.368102729434 54	3.333975740413 21	3.112091598444 8096	3.381554603147 21	3.351616536752 1595	3.1095041146294
EGCO	0.931231530127375	0.954003139270 1039	0.915732765666491	0.989377155102 0719	0.983141873887 8031	0.914899944728 792	0.937294446248 303	0.956494115364 2349	0.91556225094 9146
GLOW	1.293176894005 78	1.351566173486 61	1.244060875813 53	1.326595490026 3799	1.293825859676 35	1.262856050366 2	1.297436053137 6798	1.347321030408 01	1.24425424316281
GPSC	1.459501625196 7499	1.447518252107 7201	1.375447629626 18	1.738401036285 4498	1.718100560382 6098	1.620647376274 53	1.472361001346 22	1.445078877822 74	1.39064290107489
HMPRO	1.547915872910 26	1.586714998402 6099	1.486440343754 42	1.536466297584 61	1.509274566549 09	1.475032590428 8	1.547100727432 52	1.589707614675 61	1.48627040466529 8
INTUCH	1.627214071973830 2	1.660577006693 8	1.602663609503 5	1.661175890206 66	1.692736879720 9301	1.605642450961 3298	1.639562833540 19	1.656463717493 2901	1.60187589253498 99
IRPC	1.338544579635 1	1.405563886801 24	1.280338763771 4598	1.362316273157 2098	1.395009098965 0399	1.302281165833 1	1.340182122918 32	1.361620306328 76	1.28030288005543
IVL	2.137592611245 61	2.207432643260 8897	2.065576278740 79	2.362642069650 97	2.324620781072	2.120686404076 045	2.175051035849 45	2.231948916218 23	2.06525931244221
KBANK	1.416987141629 7	1.447099753087 92	1.406420035052 58	1.432298701607 6398	1.432864837267 48	1.429892926825 65	1.425924440783 33	1.441571359577 42	1.40651056224012
KCE	1.524397606727 34	1.516001978819 55	1.465811828049 06	1.473793171978 9399	1.455155648986 26	1.445154923048 74	1.520728452922 97	1.518788687328 07	1.46576670641762
KTB	1.207485156871 94	1.221341510006 2999	1.151957011953 66	1.195060254036 47	1.191061800164 83	1.184561912507 3699	1.202311298461 6601	1.220018145607 15	1.15136555241901
LH	1.225880094057 64	1.242773116816 71	1.203004277864 84	1.256832198820 67	1.271502012101 49	1.230761158722 18	1.224192978888 03	1.231634214284 8	1.20300409919201 01
MINT	1.644105092931 92	1.689266893700 54	1.595598802904 6	1.660557473824 37	1.646637708406 95	1.623133347238 7	1.639530926513 6	1.688121382935 54	1.59850095196074
MTLS	1.024027843738429 8	1.016176223158 68	0.941435117441 751	0.968672268649 7109	0.973072600366 8661	0.955108650929 305	1.011812852794 21	1.042324681670 29	0.94255806923366 99
PS	1.278768848850679 9	1.333371324253 2699	1.240419736496 0698	1.332935963510 12	1.357555434114 2	1.239768163129 92	1.304430971929 43	1.304395472498 599	1.23798780596341
PTT	1.918849588413 82	1.940610411539 79	1.877545387791 3202	1.822288604567 6099	1.819827470345 5	1.809926885107 9198	1.916213004924 3098	1.936330089068 29	1.87941799892053
PTTEP	2.236690144056 46	2.236994127285 498	2.207397605822 14	2.165701907168 98	2.168744815443 8402	2.173926163642 17	2.235552266618 87	2.228922006041 2	2.20740831229547
PTTGC	1.842040515372279 8	1.868086703478 58	1.776484264902 5998	1.796045954449 5402	1.778748703862 44	1.820656138471 3198	1.838959711828 9702	1.874666494933 72	1.77669155943309
ROBINS	1.487363314444 24	1.524658386203 54	1.447422295592 21	1.466798520200 74	1.480676926833 77	1.441870745758 63	1.486974097196 94	1.527796580923 05	1.44732079760801
SAWAD	1.519994427361 41	1.556383555786 482	1.408666006098 502	1.486087460940 01	1.464163629433 99	1.390457564117 21	1.519904655285 86	1.515653320199 39	1.40877086246626 98
SCB	1.469471871070240 1	1.503737671587 71	1.456869499283 92	1.476464302348 81	1.485455099601 91	1.467831369187 8	1.467208351356 86	1.505781342321 4998	1.45696078159972 99
SCC	1.098993382628 15	1.087281494913 95	1.071312727243 26	1.095936557777 4702	1.093110062976 42	1.084463533875 45	1.103773004177 29	1.087957487675 84	1.07063674253494
TASCO	2.413378673226 96	2.440434309133 95	2.337964518485 63	2.568111021398 42	2.615776820631 31	2.633390471626 54	2.410338957104 5097	2.415064534385 77	2.33775367146249
TCAP	1.200403788047 1399	1.217023072609 45	1.129449522604 86	1.147993347877 59	1.163965457620 92	1.127122366916 07	1.203726497637 4	1.213818037244 5901	1.12951566978 3601
TMB	1.487626638154 4399	1.526703070000 887	1.386083036171 16	1.515358320454 85	1.515141259434 58	1.516032518380 5798	1.510135058137 87	1.522967088585 912	1.38560248294154 99
TOP	1.558579417360 55	1.557193746751 92	1.517537453815 1498	1.577397748139 35	1.587810655344 499	1.514003950414 3	1.551740266081 001	1.557811603621 69	1.51755415772020 2
TPIPL	1.720288027931 8902	1.802587684485 13	1.592440061346 83	1.895555207152 36	1.858339684933 5898	1.768246961684 1099	1.757927898626 27	1.780010799090 02	1.59211645976227
TRUE	2.021119887055	2.075537059232 1	1.957537147000 2	2.176641032813 86	2.138328233511 31	1.999606491110 07	2.057197038717 51	2.072489558902 79	1.95773475943513
TTW	0.839069503497292 9	0.898519103082 14	0.821803902120 1061	0.807632906234 836	0.834763705085 153	0.840031003208 813	0.836993685731 954	0.900058205458 8041	0.82405479717242 8
TU	1.395784658135 3102	1.408488942167 79	1.364180778100 74	1.403200701341 73	1.411466573051 899	1.394829840457 26	1.391074550695 78	1.408245162302 02	1.36421529896 237
WHA	1.540495456359 62	1.590609842018 37	1.513143223925 49	1.603083295180 65	1.629215509154 8501	1.600889350488 59	1.568832070989 199	1.634065677895 701	1.51309836815315

PredictPrice8 : MOF (Money Flow)

Table 22 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	33.28565473766011	31.8511216760273	29.5095956644527	32.939489406761396	31.0328042579499	30.5303242676492	33.3524150416669	32.731543489714696	29.60025847111161
AOT	28.9394472444028	26.948546666321498	16.4073528932179	29.4356056823349	26.644224911278602	17.2666314184244	29.067013338392698	29.2048440600749	16.26668650028211
BA	1.0851990572641484	1.0854494788018099	0.926819895577934	1.21422445652138	1.0568415508243	0.9814613993388073	1.053656209057558	1.0545940221879702	0.9266786635271159
BANPU	36.41266246502444	33.53247492215411	16.9331927112507	39.69115834980046	33.83468873765533	20.751326819114798	37.7746524000221	36.321072663319995	17.00681058404745
BBL	6.561443783208155	5.202257053504906	2.7887746709762	6.423680224916633	5.684849610516256	3.2099253440761197	5.7858351111674661	6.387807569896111	2.791937559005277
BCP	1.585650114873571	1.581666812451551	1.422748335595481	1.6257133206929002	1.6334000546594	1.440534080333707	1.592574356634071	1.7377332599613202	1.423081705606081
BDMS	15.049548679697	14.966138633023702	6.489035933431859	14.213139142610201	13.817118275057998	6.969131343941379	13.0276286696905	12.09702860347877	6.440388794765741
BEC	40.72078275538086	38.8514156369433	17.4947713481259	39.226767484560696	35.039916073968605	18.49704159264354	38.5049652724476	34.49783351543	17.539993181328
BH	1.837331508668551	1.7120644084708099	2.037817746542994	1.7458172218081598	1.6389633975555502	2.144119913703447	1.9046884537817301	1.7217520949227298	2.037178691134744
BLA	18.14872428956603	18.96669453323543	7.841331170280901	19.76678628866554	18.08646423213559	9.433633353284708	17.379575307274 3	18.2061480414109	7.765905434005505
BTS	0.991988746134446	1.109552188338875	0.989294684296093	1.0681946548163699	1.0833319436933	1.0207514639309598	0.993104508457518	1.0646614181775	0.9903213468683459
CBG	28.3367495177188	28.188413910654596	26.541604457221403	28.38159371284 38	28.265417519275697	26.696638181858 24	28.114702849266298	28.1905440982555	26.402891740845803
CENTEL	7.003755317660230 5	7.193922142918	4.155237616786 64	7.499687526424 43	7.43348386515 07	4.796125910159 61	7.314554822889121	7.049490911575 49	4.133954658936
CK	4.364666636255 74	4.788975861434929	2.539633973933 96	3.987485003981160 2	4.641581553898 98	2.682658301769 58	4.47949129921591	4.342792135797 32	2.516762717378 4
CPALL	2.759502598524 28	2.756918088020 46	1.897922365733 48	3.053517574994599 7	2.7859912928924	2.042963282789269 6	2.751796019607 92	2.721608226557 73	1.896999424682 03
CPF	4.704998315412389	5.009486383594 85	3.511594688758296	6.440930284794461	5.190996444772 19	3.414490195375279 8	5.258740965520229 5	4.757488690645 52	3.527147142208929 8
CPN	10.53056921765 42	10.802181249686 1	7.629734452174 88	10.910110577346 6	10.079121812392	8.743363328833 03	10.707376855633	10.573467612158	7.577424588013 711
DELTA	1.982001012321 96	2.023167307864 69	1.885861020959 49	2.129806396034 82	2.033057289449 05	1.933295412809 34	2.018738996286 41	2.038949734591 05	1.888460901594 801
DTAC	130.0963451575 62	129.778681708 424	65.648466321133 9	134.123654371145	120.208148336919	73.902643009916	132.873070082953	121.053916633004	65.99762120491 66
EGCO	8.269480775237 03	8.614801234915 05	4.484936426424 82	9.354964615883 32	7.761304460654 03	4.628197314532 01	8.482613748359 92	8.941385730296 709	4.476027666216 22
GLOW	1.289029035195 29	1.363764327286 1	1.312783432850 98	1.394487935028 23	1.471331667533 36	1.376314408023 45	1.277969318076 07	1.337294627480 301	1.312187225564
GPSC	6.447103513493781	8.136598017142 17	2.220740256349 4796	7.157291023944 15	6.651319453559 8394	2.894189732100 34	5.542030952822 25	5.716405957767 78	2.279947800565 97
HMPRO	5.776202618347559	5.291542331606 365	3.485911581101 29	5.558274557626 39	5.343262147443 24	3.541766053295 3303	5.415856612972 85	4.776365413279 67	3.489040757007 692
INTUCH	28.458426501099 97	27.930910225541 3	27.323061357283 603	28.440868758237 6	27.539270948511 003	28.610903610859 502	28.708955820139	28.125557320328	27.459190651548 5
IRPC	9.429450869091 82	9.857801727602 47	5.408287742399 23	11.936587113387 2	8.869878449604 62	5.886796826011 7305	10.699417631176 8	8.704203642582 52	5.441776262442 069
IVL	8.67331925414 29	7.589209037383 469	5.610072592468 86	8.637706165894 24	8.047521331636 945	6.270275062434 54	8.772402942413 31	7.389875024031 54	5.596315579113 5994
KBANK	2.296759066961 972	2.582401263653 15	2.134493503171 69	2.303671827890 8097	3.372088617660 0398	2.300624352411 01	2.273472646683 66	2.650524397595 39	2.135956717961 8
KCE	41.426590984157 37	37.384997049195	24.591887221597	39.280527230078 4	38.300111608183	25.601214991031 4	39.416369351376 0005	39.408858006237 104	24.466147781844
KTB	1.369973373195 58	1.598532359686 4098	1.223372327842 7902	1.646316218389 8001	1.711372593193 39	1.275096287062 83	1.360244610840 15	1.843642659609 64	1.229507128533 79
LH	1.360197076428 62	1.417066251413 35	1.326897832551 93	1.558077223468 92	1.416581069889 91	1.304016736210 38	1.379676115884 598	1.371612061313 97	1.325457212566 17
MINT	19.650230951672 8	19.212925891345 7	10.698838420284 2	20.192092572006 65	22.534340294736 8	11.522291354571 7	20.433259277724 9	21.869730243389 1	10.576279998334 38
MTLS	3.807391471213519 7	3.865409109883 46	2.589608074904 99	3.964086976346 3	3.367494921471 85	2.575187400414 06	3.755929815591 82	3.719937384370 202	2.585644867561 61
PS	1.284204336020 53	1.421657611770 159	1.431230326333 27	1.353962290314 76	1.455394399069 39	1.480706092824 8	1.354294388592 81	1.432578480810 14	1.427387136304 63
PTT	6.542206142724 98	5.803983049394 62	3.369269230538 7304	5.647031755072 87	7.545646977923 8695	3.712192527939 4596	6.434109778008 84	5.950479483288 48	3.392800055696 65
PTTEP	24.746461151482 197	28.156457943695 003	10.378766587595	32.531459440774 604	33.77302571249 93	10.620890617135 59	26.401338770236 197	29.129445018495 296	10.416724230425 501
PTTGC	3.969665776000 1197	3.772274170451 304	2.664258576202 6503	4.583349086741 7	5.46578769737 99	3.009589612988 75	4.041640349134 58	4.496143077456 62	2.662947571427 3
ROBINS	7.273224368315 74	5.588395738545 33	3.263242623404 5804	5.741481002666 58	5.465781833014	3.728611683460 84	6.782009749438 8305	5.858596306850 46	3.260482327380 01
SAWAD	1.914845509944 32	1.898233007069 288	1.660154545504 325	2.108355855871 54	1.943330637183 0001	1.648331743948 31	1.898419505170 55	1.805524668000 98	1.658920855872 2439
SCB	9.242386379014 16	11.936592722603 8	4.609526667582 28	10.340034966246 5	9.300796226746 9	4.779182510438 41	9.736142194325 43	8.614587502048 26	4.620976795618 479
SCC	1.378175180827 63	1.536154601058 7	1.198963288503 11	1.393114434933 9	1.384430749760 8601	1.211213501182 07	1.318666487964 48	1.446941500324 97	1.198632321855 2501
TASCO	23.250098094369 9	25.258836724749 97	12.303191972684 2	23.728095815455 998	24.036993176433 2	12.873998498881 8	24.852843226886 4	26.330825372795 502	13.00270302850 82
TCAP	3.089399985406 74	3.084188526069 72	1.914541382430 92	3.527193495712 64	3.315579643009 24	2.218178893036 7303	3.128028192078 88	2.989435596416 47	1.913731532937 95
TMB	1.615301925581 79	1.711382047583 1	1.145648022574 123	1.666685020394 0102	1.680732960238 45	1.491621359166 44	1.721489789371 5502	1.680066005798 199	1.456754337626 33
TOP	21.191553730571 798	20.369040256588 4	13.872550603777 6	21.055225365 415	21.153114773462 4	15.035855305708 502	20.814259681351	21.073256381901	13.648582027376 4
TPIPL	1.914918244428 33	2.062531327949 4	2.356643556790 22	2.136077303116 89	2.103875348259 79	2.349755864422 94	1.905359810217 52	2.259052897246 48	2.354347586430 84
TRUE	3.682334537711204	3.174411195843	2.841598555059 31	3.030719376701 98	2.940318212806 63	2.865077656216 87	4.558289906054 9	2.835953717726 22	2.829708645441 06
TTW	1.122316772314 5	1.484332680855	0.994503875498 5921	1.318940454024 14	1.513886145796 64	1.019796324949 2001	1.042090556477 1998	1.559106372979 26	0.999351322365 717
TU	1.499383368931 97	1.576260829345 37	1.327049148405 86	1.506822302831 44	1.539249545275 3099	1.375353168124 98	1.560909839933 45	1.525497531027 69	1.327466148846
WHA	1.688904806931030 2	1.824375385302 3	1.670874637617 99	1.912804714913 14	1.896223540765 1702	1.822834572346 61	1.641653370493 6402	1.831616517358 46	1.670933904042 67

Table 23 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.900506052535470 2	1.83505732579387	1.793216121093741	1.836254214995 26	1.814486973660 94	1.783641380159 87	1.900419866906 8901	1.842117637281 35	1.793173446140 65
AOT	1.040040142895 83	1.047838390871 0098	1.037553300478 85	1.059212140918 55	1.048893861025 36	1.046012520980 8702	1.036521523037 47	1.045500746285 5	1.037608951831 67
BA	1.040585240422 02	0.989031394087 9379	0.936268022598 7491	1.072399446461 57	1.035573169881 66	0.951197694981 417	1.053911960078 55	0.975798062627 376	0.938321106675 285
BANPU	2.027782605636 61	2.100212972686 2	1.999171001084 64	2.088126545315 35	2.087244136855 0003	2.024417488581 13	2.043293997473 38	2.101669141958 397	1.998657414155 26
BBL	1.216739519486 54	1.252105957058 09	1.216433286931 9099	1.253006958709 09	1.238680193690 74	1.227623571373 5901	1.237613867583 12	1.257236495865 71	1.216155271620 54
BCP	1.350714296883 55	1.378582600349 2	1.332875855069 21	1.325260312111 3	1.384439187043 74	1.302570806061 7	1.334853656088 39	1.384340371316 09	1.332688942501 87
BDMS	0.925286218002 588	0.947036349191 616	0.892865879183 41	0.945003103217 4699	0.939318900018 5379	0.906107957951 781	0.926188148908 117	0.953259001173 544	0.893006604640 061
BEC	1.939399697215 1402	1.906805950962 5899	1.714089990215 098	1.918707124835 96	1.827066613619 198	1.681486732388 81	1.941956602660 702	1.903357337362 6302	1.713396322705 14
BH	1.460882984007 6398	1.467686320774 871	1.405821783350 87	1.544005770611 24	1.511305548478 59	1.462873785955 31	1.466419849234 49	1.465952957494 86	1.404789570134 67
BLA	1.684586303604 8898	1.681242510736 6797	1.641012292837 326	1.824920030375 7	1.808132856964 89	1.730487600622 1	1.674328948433 3	1.678997281147 19	1.640799733375 49
BTS	0.927229253699 885	0.986508585876 3389	0.926777162591 188	0.970691429402 264	0.985162102662 801	0.986489481137 445	0.936748499371 738	0.994637930938 6929	0.926438186066 162
CBG	2.039068780219 604	2.001675129658 69	1.965407054870 7998	2.03719194215 71	2.029451443734	2.050441788754 19	2.033245956624 0804	2.051817752798 6	1.964658493317 25
CENTEL	1.629062173289 2099	1.672711111436 84	1.513046516379 97	1.603756403523 83	1.592776350823 41	1.528909145108 11	1.626956134499 9999	1.654389971694 799	1.513146198658 15
CK	1.401643925305 9298	1.438638359564 01	1.355625414971 4402	1.384251449245 38	1.388987667043 22	1.352400995796 202	1.407641607318 29	1.431305265964 86	1.355052766168 09
CPALL	1.351297406313 29	1.332911509834 02	1.320469901426 3699	1.352424726276 94	1.355514425204 3299	1.348246209201 49	1.348282835488 65	1.342972957039 2899	1.320000411434 03
CPF	2.037248117476 18	2.043775255416 7	1.987552807697 41	2.026015838799 26	2.009465718610 12	1.987561341793 71	2.043332893475 47	2.046521237407 94	1.987394328758 1401
CPN	1.306707158470 83	1.296772244317 43	1.261141489980 64	1.228771346080 37	1.232863777613 03	1.213426519756 5002	1.302484341212 91	1.296750026019 399	1.261349548328 02
DELTA	1.724922562627 55	1.732819687201 96	1.705904714003 0599	1.706904889460 698	1.708945705578 43	1.712435140000 59	1.742849533214 299	1.735677183329 55	1.706207428854 6902
DTAC	3.213431894932 31	3.378556088483 14	3.097367350343 4203	3.336331176288 0703	3.466866397083 6	3.159014550226 1504	3.175172908084 05	3.319655736950 96	3.096748784021 16
EGCO	0.940811830347 531	0.940236795679 052	0.920866378176 656	0.918465818847 787	0.924345650878 244	0.908138548213 927	0.950909966507 534	0.939931862136 738	0.918894946492 123
GLOW	1.325460507258 84	1.405516592208 3002	1.251606384267 17	1.328278878257 5	1.322568939343 223	1.255293341718 89	1.319843071908 68	1.389484147864 08	1.251509759324 72
GPSC	1.410371661036 9301	1.426524100343 36	1.568476247415 3398	1.798747359559 4299	1.771996822161 38	1.858745151847 999	1.471924732490 1101	1.429451011655 87	1.554546147646 74
HMPRO	1.479802644583 43	1.503398660684 16	1.469702783850 54	1.490855164679 5098	1.478303730933 8398	1.459830296035 7599	1.467814192377 9702	1.511585680422 498	1.469703892860 14
INTUCH	1.676902726303 45	1.707510838683	1.627802546081 9099	1.603464380245 8999	1.640085761087 56	1.621316986045 7102	1.675662241508 72	1.706498819040 1298	1.624773484167 32
IRPC	1.322628554657 51	1.340195607148 91	1.288699382987 0301	1.318618301966 1801	1.312182010456 91	1.293178215841 402	1.309160861839 64	1.358345592063 8999	1.288327892036 83
IVL	2.163149229689	2.169578240525 91	2.091679098898 47	2.236496453270 81	2.221998892370 4004	2.149418335369 9103	2.161629362280 63	2.160109079626 59	2.091836173006 25
KBANK	1.364346891079 12	1.381293046511 47	1.382343395845 28	1.361441508459 799	1.360831333623 15	1.418850641374 46	1.362080090461 64	1.390259386113 37	1.382520306974 45
KCE	1.472043630548 57	1.497495119133 26	1.444846463464 65	1.487270735089 6301	1.485899976141 31	1.456142004375 71	1.469032487866 23	1.495000913239 0398	1.444903590719 42
KTB	1.162090542452 62	1.204427722086 7299	1.145214280297 501	1.205861734470 04	1.201624602807 69	1.191883028614 24	1.163587259253	1.206109561867 91	1.145296876111 581
LH	1.262956031459 31	1.279338135168 99	1.206026472357 93	1.260659584928 65	1.233470014003 3799	1.203242352234 94	1.261488636860 987	1.277024877670 8	1.206009346877 71
MINT	1.641501055715	1.644857569835 69	1.592230410092 3899	1.618684865709 56	1.600655630889 74	1.593585603866 94	1.644719482337	1.638484265528 9	1.591211822355
MTLS	1.047831827496 6	1.068345598290 31	0.967848034923 258	1.121644842515 402	1.041818780789 736	0.946021922524 465	1.063193887849 44	1.057757646947 34	0.968591039248 7909
PS	1.292046285984 95	1.334472566167 68	1.252384302303 77	1.313881154088 799	1.321236720597 51	1.240050484577 9	1.313153811178 67	1.338196832357 19	1.252995851858 92
PTT	1.842524148323 64	1.888449935089 7501	1.857566876541 4	1.837242752363 16	1.837115780795 8198	1.842204368604 43	1.842770078076 98	1.894040258136 5399	1.857754642266 65
PTTEP	2.389957174730 36	2.429505949817 69	2.335320356952 71	2.274890004854 2997	2.272843513813 6103	2.214502357463 56	2.406274229364 6	2.420777745251 55	2.334977512715 6502
PTTGC	1.777177761964 1	1.779456538653 96	1.767028407513 6398	1.781694129767 14	1.768814352032 98	1.793782559197 87	1.772540034849 72	1.805309376120 74	1.767168161584 2
ROBINS	1.452705341836 23	1.478814810585 073	1.449090236240 9301	1.447999016363 72	1.456840723914 62	1.448216245953 9099	1.452011314550 6501	1.468200267686 12	1.449026036384 12
SAWAD	1.482569311395 15	1.563881977654	1.405992251896 28	1.507722050149 7501	1.472131697497 97	1.423972071096 62	1.480551127097 31	1.538061289378 92	1.405678841911 099
SCB	1.395775217224 88	1.413377444074 874	1.431988237484 9	1.469612194183 701	1.458513793308 46	1.447088986009 72	1.398270655146 01	1.412748896025 15	1.432181676200 55
SCC	1.127787530366 1	1.117958351756 21	1.073365105317 89	1.114034614142 21	1.109382608755 9	1.087895775901	1.127681438567 24	1.119146897474 61	1.073140526428 1
TASCO	2.392477020644 54	2.399169001605 7903	2.338061172793 13	2.525164078504 51	2.517385393691 66	2.506085236376 61	2.398814418169 1903	2.398121402234 23	2.336908167956 35
TCAP	1.185042513748 4	1.183336302212 7599	1.123575238351 92	1.191102874597 84	1.156835998246 8202	1.129533128609 84	1.184615211336 9	1.181719698100 93	1.123785434043 131
TMB	1.400928850746 12	1.411079282099 651	1.391647571572 62	1.498486341981 82	1.480283703027 49	1.516416416596 47	1.399473171986 2	1.411540280184 03	1.391687897996 58
TOP	1.540604737917 14	1.546292517877 3699	1.510098677310 25	1.540509054103 699	1.542235124559 07	1.509037192972 27	1.534249097788 25	1.543080323391 61	1.510604241004 47
TPIPL	1.614929723065 0401	1.768106445050 9	1.605963896397 43	1.868896764761 7	1.870721962258 11	1.740724885436 55	1.609796789603 55	1.797305934218 42	1.605614658991 3198
TRUE	2.062913557930 2402	2.147291837432 103	1.977339238749 19	2.125396003607 62	2.158099699009 87	2.051495283187 503	2.076004943922 6	2.151860549326 15	1.976514516232 1
TTW	0.823619743685 897	0.841197015895 141	0.820893748511 534	0.863852894414 722	0.869023812959 5909	0.830613802890 298	0.832748416759 396	0.840972886672 619	0.821145040326 247
TU	1.339025015010 7201	1.356408510571 62	1.317817162001 04	1.387343203383 16	1.392324518761 06	1.351771435766 64	1.325493582802 501	1.348325057251 54	1.317670345744 57
WHA	1.583453028535 1403	1.629404397409 3901	1.535632110566 1799	1.677562986005 1398	1.706302774010 45	1.579134993758 7	1.589982177133 65	1.623500895124 6501	1.535758401164 31

Table 24 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.80131033387201	1.78198136025685	1.7685707365980199	1.76210340915415	1.7531055538155298	1.7404691032389101	1.7938058590387398	1.78187777210559	1.76809375440982
AOT	1.0373841475652699	1.06268353918201	1.03322777759577	1.0904892941645	1.08255667323897	1.02929971212136	1.03292211968014	1.0830715263370099	1.03385450942917
BA	1.06092929243976	1.00727806421161	0.929644761723695	1.05101560616771	1.0097564443965998	0.935421092290599	1.06359980252743	1.02531151597591	0.929620558907901
BANPU	2.04373236082234	2.10251754903489	2.00701227687277	2.04411083123098	2.0589600216494	2.03461782294052	2.0529997101711097	2.11089502983503	2.00664759478769
BBL	1.22217874641569	1.22364594467258	1.216189159364	1.2477280077175599	1.2390735368526	1.22616579001579	1.2218226793403901	1.23151600672868	1.21545032096696
BCP	1.34918923386355	1.37890536238381	1.33287585056921	1.3260095036203101	1.36954304717545	1.3025708060617	1.3515682075884101	1.39106978723003	1.33268894250187
BDMS	0.9411694739197901	0.964237055205759	0.8928658791828961	0.9509779944644031	0.9459203921586079	0.906079465557657	0.9301883299504741	0.962956572217512	0.8930066046398171
BEC	1.81035736999798	1.76973497385465	1.67838843990398	1.84498313150343	1.79525007067072	1.69174318516397	1.8114583986499102	1.7758168099134102	1.6784035033065199
BH	1.50052099014879	1.524013369502	1.4104820865137	1.55123304181129	1.51394812815691	1.5256856007699902	1.50029660846143	1.52265827068867	1.41039210644857
BLA	1.6806457893762599	1.68692379473506	1.64086527166384	1.81372237699306	1.8082626826242498	1.73028784869201	1.68463573200658	1.6814375933993102	1.64079973337549
BTS	0.9249550581869791	0.951737280630335	0.913435793590551	0.941330697520428	0.9540627297895079	0.97347563272217	0.924201039817996	0.9489760940020671	0.913512109168336
CBG	2.08366409046939	2.05107298993768	1.9654070548707998	2.0359442187847	2.0487788847496398	2.05044178875419	2.061792952495	2.04043962320737	1.96465849331725
CENTEL	1.6455503285526099	1.6869767001698899	1.51393046546619	1.60923105693719	1.60692869416145	1.5323202256583899	1.65210693180551	1.6751429122683001	1.51388227405669
CK	1.37014049500694	1.44318984431277	1.3612286394623	1.38907824508812	1.4321399393866299	1.3447524129464599	1.37423463189974	1.44335916955658	1.36053296206992
CPALL	1.34920977692843	1.33582412733934	1.3204699014263699	1.3500177064646501	1.35109126514297	1.34824620920149	1.34764430139981	1.35148039462613	1.32000041143402
CPF	2.02186315247455	2.08211546361117	1.98424722877103	2.04131436579063	2.0396374340806	1.99993745291521	2.0096529618020402	2.0465381783466596	1.9843680364926501
CPN	1.3028550655172	1.29555273757736	1.26114148998064	1.22987944828227	1.23276886154618	1.2134265197565002	1.3071216430008499	1.28871789199213	1.26134954838021
DELTA	1.74051555311111	1.73140441432777	1.7059047140030599	1.69180530845894	1.72021649314861	1.71243514000059	1.74144419224868	1.73189349847007	1.7064794641717802
DTAC	3.1046412023724	3.21135829789399	3.10038159096237	3.15342992290696	3.22971056497681	3.1738516313114395	3.13217814064504	3.28793233213732	3.10042691466666
EGCO	0.9517457089576371	0.9393306445114029	0.9218973776013081	0.920209222928557	0.927013488917344	0.908616014438793	0.952693136751868	0.940838265585471	0.919873597187749
GLOW	1.34052018604009	1.3801698217645	1.25130823740402	1.3251629741317	1.31893337793354	1.25606866319176	1.32718846004633	1.3768299565419	1.251360116284
GPSC	1.44677007446411	1.4323182392151799	1.5684762474153398	1.7742497291766302	1.7905406344494001	1.8587451511847999	1.4093582514202598	1.4282648443969301	1.55454614764674
HMPRO	1.48662686380379	1.49823830645084	1.4699244024504199	1.49526294232188	1.47076003785059	1.4608969110527699	1.49379990913111	1.51752805254983	1.47004638543693
INTUCH	1.67093724159415	1.70746225913976	1.6278025460819099	1.60609430495279	1.6428336779494699	1.6213169860457102	1.6756789042735702	1.70772787667542	1.62477348416732
IRPC	1.3173382337725799	1.36404572858587	1.29015226249643	1.32174780225905	1.35905866333432	1.2880499737065798	1.3115203000283702	1.35112588259118	1.2895470445575798
IVL	2.19670095619964	2.18949836262794	2.07849278418945	2.25752123377459	2.22211820250751	2.15641802884369	2.17405973491224	2.1894785190194996	2.0772390309835203
KBANK	1.42653565345559	1.43380107168487	1.39201080564387	1.40680303949605	1.40852806611009	1.41670588780286	1.4383590853661	1.42174345384334	1.39179230772554
KCE	1.48117649032357	1.5056574483379401	1.4531458811179498	1.48426371367772	1.48870472571236	1.4603495024025601	1.49659507098045	1.5088991688434399	1.45307227574791
KTB	1.1624453339943301	1.21411624242164	1.14522877986315	1.2027879968464	1.20199193544263	1.19189347262347	1.16214007158339	1.21009708028287	1.1452968761158
LH	1.2662701333006299	1.31159746886706	1.2236424100407	1.24870802911632	1.24916967735599	1.24226728027647	1.25974282534744	1.29922270474186	1.2238647512738101
MINT	1.6116943944531201	1.63469439619147	1.59594442847731	1.6354442260546598	1.6679526050810503	1.6046286107592602	1.63172691147856	1.6349767857069801	1.59594455034625
MTLS	1.04952630973009	1.0396612122150701	0.967848034923262	1.12088053839778	1.05341379092848	0.94601322418764	1.0488442652816499	1.06639755384002	0.968591039248789
PS	1.30858576570666	1.4036634581843799	1.2823193755638	1.29388792469705	1.32227492784795	1.2063461611153399	1.32573430121616	1.40559654509625	1.2809404461912701
PTT	1.83473551965445	1.8860979335968202	1.85988173999914	1.8290323722640298	1.8344211953503098	1.8237872527038699	1.8393231051294399	1.8903544508191001	1.86006018089303
PTTEP	2.23148783711826	2.23039243989599	2.24465513406656	2.2061154803479	2.16811518064869	2.22180031517366	2.23288970970517	2.23011021278974	2.24372349811545
PTTGC	1.78724977910402	1.8002066320892698	1.7670284075136398	1.78135448878388	1.7708587729948	1.79378259519877	1.7721718511718598	1.78251306788128	1.7671681658142
ROBINS	1.45450356762485	1.46571003290618	1.4490902362409301	1.44968694072664	1.45824587332629	1.4482162459539099	1.45616228395661	1.4561935823277399	1.44902603638412
SAWAD	1.5160185670314401	1.5569177097483	1.40599225189629	1.4941863366230899	1.4701021070384601	1.42402065051901	1.44850693834692	1.54407936468556	1.40592185070968
SCB	1.39253579348286	1.4128780225322202	1.43198823748489	1.47024236074531	1.45826024584119	1.44708898600972	1.39950924091392	1.4122623456153	1.4321688757499
SCC	1.13102492361818	1.1143924852808	1.07551002078758	1.11328713489018	1.1135098300305502	1.08665216977437	1.1320196193875	1.11226289852371	1.07563959257539
TASCO	2.37096182732928	2.42072283079437	2.34870070120539	2.6059513432332997	2.67522787800624	2.65208079840111	2.3683363683070597	2.39934700965258	2.34920883672925
TCAP	1.1837894045208301	1.18405297215543	1.12357528385192	1.19279515600484	1.1528986593039798	1.12953312860984	1.18431925765796	1.18132133465531	1.12376607059458
TMB	1.3811038313950201	1.41497469929236	1.38706819601332	1.50940931894986	1.4685127279343801	1.51501161486583	1.38818187848977	1.4161608763142	1.38697166074225
TOP	1.54019422751418	1.54802283960964	1.51009867731025	1.5405017571569901	1.53955166877088	1.50903719297227	1.53037189928959	1.54546245477992	1.51060424100446
TPIPL	1.60804729335013	1.84335442347296	1.6211441795044201	1.9215803755299201	1.88876910206211	1.76127942734563	1.6516982056264702	1.73927045835319	1.62080468748334
TRUE	2.05087460218335	2.09005822571961	1.9669367817528298	2.07985626527112	2.07556264237798	1.9773527666955701	2.0790685035300402	2.07788384059718	1.9669794795744
TTW	0.8378730253316421	0.854134598354204	0.8271564785353409	0.893604135534116	0.8770098292948639	0.819537560861491	0.8604359011966021	0.854749038053167	0.827230286919939
TU	1.3334185338770101	1.34849768132941	1.31779563781649	1.38138348199888	1.40506153937151	1.3517136780786598	1.33642198830247	1.35137188436781	1.31766625094584
WHA	1.5808855522922	1.5990436309978402	1.53341511157234	1.6909043678682099	1.7100741683549299	1.5934364849030098	1.58527608786656	1.64489772496133	1.53328958238571

PredictPrice9 : OBV (On Balance Volume)

Table 25 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	31.0208123497068	31.3229355181363	26.698239083193698	30.568070072487203	30.855362535263797	27.740910859952	30.901577341135	31.2431751539486	27.1622017179359
AOT	27.6672357804072	28.203082660233502	12.7977331014897	27.6531381012182	28.043466066468497	12.907387166182598	27.674622230755002	28.169349741306	12.5888069044504
BA	3.26015409591356	2.9077970827737	1.7300490483368298	2.9191486434681	3.36866900742714	1.24166200922819	3.25600433030024	2.55559899947999	1.74941517626405
BANPU	48.536894807898	48.7887882445393	20.905553299405	49.4226589539373	46.2702929120882	24.2589123792757	50.09527801896711	45.9976115821342	21.030927521387298
BBL	10.7019995215464	12.2918459831244	3.45706702918603	11.1315558012229	11.6952088937749	3.7351789324983398	10.6987245202984	10.6294817930299	3.4109360156064796
BCP	6.440937666663241	6.5442733582945	2.88922645558067	6.07476766513759	6.16150990618487	3.13741401077166	6.19570755842176	6.24635509109606	2.89268415209783
BDMS	19.5620431875063	20.0358632021163	6.77982512075066	19.5693379119949	19.8814901529574	6.981626436246249	19.4684004566829	20.002141945899197	6.565968044827219
BEC	49.4294639636474	47.2836477097774	13.8798621096633	48.6020597354972	48.606083808446996	15.8129265101865	47.4974214694313	50.5280018780187	14.1625908416901
BH	4.67775891715857	6.4391562765163295	1.70131536245227	3.9431111924573403	5.89198511899666	1.6671769219578698	3.031991279319	8.09969371162531	1.6852290333735702
BLA	22.667604545066	21.465479541975398	9.455945977429291	22.6354603807931	23.028486064459898	10.2098815764454	21.2409990190462	19.7142631416231	9.342736922515831
BTS	2.0413686488567397	1.8525757960779599	0.996128200386983	2.14335711683461	2.68065429667284	1.05938501809941	2.39730226909349	2.52829649525505	0.996181537406427
CBG	27.4972702820738	27.5902202938717	25.515658924669598	27.253873945210298	27.666652417055797	26.5315344934703	27.594953934758	27.7584509513898	25.5099051228794
CENTEL	10.2744277110402	10.9888502735841	3.58047497479428	10.428339698641501	10.145776952470701	3.8821617022764503	9.47194624956955	10.182490486840399	3.51784485989362
CK	4.26245696300696	4.63896507942558	2.15510584766161	4.4517675250926	4.9688478233587	2.31749824730873	4.24542474370183	4.7697384489168	2.16987819928427
CPALL	3.15046238352426	3.9775450030801998	1.6161819609679902	3.66392480453659	3.7264026060372304	1.90707739720002	3.42204849622319	3.9053398544221	1.6134406544110198
CPF	16.0296329610462	15.030594209863299	4.026323577158919	15.4516810685355	18.1765195893479	3.95059637138937	16.2492950710797	18.2421054675143	4.07855642875562
CPN	10.2449069116634	10.4549125682932	7.98730225648345	10.4163648824721	10.3536582368382	8.67478338764708	10.1828645927052	10.437543840828098	7.95647329675358
DELTA	4.05370696186198	4.1815940263669	3.3774584109021197	3.57529690548802	5.43076101662664	3.63676359186016	3.70340515462488	5.75029110312726	3.3052088387773395
DTAC	147.209453271677	153.676863094955	70.66757832718031	148.45424778277	156.752780283087	71.855671731336	149.63269941525502	154.447293721221	71.8523862618765
EGCO	9.304775901487861	9.76198638147527	3.61990559149243	9.30428065472223	9.67521614295275	3.5969776131330495	9.09415975852014	9.694176360669731	3.6054383372950403
GLOW	1.76078504350776	1.6651402564671898	1.5056106468672699	2.1779212534782997	1.8376430591358202	1.33332427603293	1.7613486849059299	1.65579620044065	1.4991275547090699
GPSC	4.61810760204281	4.91867203839683	2.53422346302321	4.80580742208853	6.990969367677891	2.77948787941718	6.14480643038959	9.084826708714889	2.65250027044564
HMPRO	9.87623280393434	9.533791636830049	3.26970069443729	11.2337393808724	9.40622139528174	3.53856267042565	9.98114539814313	9.989986628459251	3.2610178052128096
INTUCH	30.262026348952496	29.9638546118328	25.9939998232948	29.726714108577003	29.8194667958988	26.896061141514	30.2037295538813	30.100271601415102	26.1583091847315
IRPC	10.7217908915731	10.9367117559513	3.83379445133091	10.1831189037452	10.867275697360899	4.6274865030514905	12.057140437061301	8.87079224028374	3.98231094904833
IVL	12.9250234561171	12.7874156316622	7.37709517190863	13.375858235586598	12.584026047105102	8.30029284658681	12.9117863368063	13.0870765794749	7.33130891156312
KBANK	8.53709886938409	9.44975696768676	2.3452473177893998	8.98152649158078	11.751648617167	2.60482876041493	7.86761169663636	9.377027774260691	2.34736675776958
KCE	35.9576718039593	36.417739999653996	19.505811009132103	35.930405223996495	36.341124077862204	20.1381052212002	36.2076144277613	36.5102313032889	19.425727909814
KTB	5.01696210302568	6.28031866642027	1.2198576634746499	4.09122742031307	5.86267209708169	1.28184134470192	5.0053048985504205	7.464790667397191	1.22550516453079
LH	6.92833784082238	6.04614074290205	1.5284758792159099	6.06001796992669	5.44230812212517	1.41773907497146	5.68212983399289	6.9282093302975705	1.50483604275992
MINT	20.3569659498	20.8618564142721	8.495142734536639	20.4898160893115	20.8468076633558	10.6602509589776	20.2716524517453	20.8588863758003	8.309833036571309
MTLS	3.3402743401152	3.8156867173992803	2.97913898784048	3.9981973426076403	3.79740175550816	2.86913127286809	3.7398876483079198	3.7034998799494296	2.98125140113425
PS	7.30740856582975	7.39214031336986	1.26927402243966	8.16018721256825	7.58383035488276	1.4125012065455	7.20471560190013	8.1673675393087	1.28103711965363
PTT	13.716940071954001	15.739117170738101	3.2755416986560397	14.0031459238594	14.7629231769908	3.96891526146511	13.9657944574185	15.625732825364699	3.29577290040853
PTTEP	71.17039042198931	74.3861822790806	20.4383486816889	67.86989957509691	73.7156105468004	19.5357168614855	71.4611436462551	73.3770697059951	21.2692426642519
PTTGC	6.27910455677437	6.7607339973783	2.37706916412533	6.12281049859969	6.06634983754619	2.59830225711749	6.368254221719501	5.89765138339842	2.3776256262596
ROBINS	10.7070314620931	9.447673618123211	4.22110062133658	9.735314171742269	8.70744643887367	4.75009005442428	9.6676319416149	9.73274385480184	4.221565428823229
SAWAD	1.82894553560225	1.88517967492318	1.5913918702036602	1.90026770669681	1.84318900139375	1.68909798543827	2.0407841106097098	1.91825527931122	1.5903476450118401
SCB	17.487083498866298	16.6489042199697	4.84305497093626	16.368886674919	17.8355456254773	4.99741962711999	16.4327489938456	17.806016686069302	4.87031979743072
SCC	4.28981346385595	3.97725602043943	1.41518632583886	4.09226362142536	4.33242922955686	1.31908474443867	4.26304638595986	4.50507818246081	1.4133090830256398
TASCO	12.3000481704515	30.8550316711924	3.22874794572722	9.95162825153048	17.111008153603997	2.80008958745986	12.8537049607535	16.2105534973359	3.36839048125728
TCAP	5.95645846259581	4.48341345107435	2.33578041620583	5.6407318811347	5.54946010643402	2.42378395169187	6.22254431742674	5.18318461456967	2.3533319908791603
TMB	2.417304353758	4.18312730885805	1.45162386551265	2.35061900435333	3.5180971082477197	1.4341881809985702	2.6642648719313	4.41844965512579	1.4506538436661
TOP	18.6520850272661	18.5867090122815	10.7444404388987	18.414920206970002	18.5536821893295	12.841536533140498	18.6844350084493	18.5395520871383	10.693720508845999
TPIPL	7.84628537659771	7.88588204390201	2.17686412434595	7.05222867149959	7.902002861134309	2.42944981841628	7.84959265172507	7.50523983397437	2.13869585585158
TRUE	14.627209879018299	29.564424554166703	2.39501118188505	16.6990012731795	22.3829129843021	2.25124844009911	16.4391506646346	29.2586227172807	2.37260716795675
TTW	4.13512280512502	4.5938892142622	1.05080451869099	3.9218642230878205	4.91644929636153	1.4747634004929	3.8836127375088703	4.9429495287267695	1.0528553208448401
TU	1.37727470711723	1.6451005263770702	1.36399154823129	1.6209399497869499	1.9755141176960498	1.40372855515915	1.66675476066458	1.8128971076041702	1.3629986354628598
WHA	4.69723044025577	4.43680256784171	1.5392861582792	4.49146410760191	5.5082763834644	1.7341525922668102	4.48303815190682	4.64257488008142	1.53739501404011

Table 26 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.8088768934100399	1.8231729942209798	1.7229091983520202	1.77914904578116	1.754536619817630 2	1.776869907805 11	1.8091533647037201	1.80105435946771	1.7240261879077199
AOT	1.16270663630727	1.13348332788755	1.1087416402172	1.06388514944676	1.05656708687794	1.05908958638296	1.166571622622336	1.1315855465109599	1.108035224601 64
BA	1.0914915842258601	1.07045887864467	0.9017502789848 85	1.0709074008333799	1.0309597121479501	1.048393684113 98	1.093375034363155	1.06102718429251	0.9046467040728 01
BANPU	2.17130424759464	2.12537333000765	2.04628478547139	2.1005758326752497	2.10761735030824	2.1846674093347898	2.18743163510646 98	2.1348811284105302	2.0469798238821 1
BBL	1.24240398437841	1.26151243710877	1.2280280261149599	1.2845118904654	1.29794758321587	1.23867814752603	1.2388052121859001	1.24155302865761	1.2278402139081 3
BCP	1.32515217653085	1.35047126598622	1.33540051863907	1.40626574729515	1.39143135700175	1.35075503151811	1.32679727170819	1.3585291131816	1.3359868707213 6
BDMS	1.01020310537104	1.02840099881289	1.03408072248782	0.9462437343801681	0.9293928415462629	0.9542253364593691	1.01584946270263	1.02817051659015	1.0304709316298 99
BEC	1.7946048057203599	1.761958663738 83	1.962395610626 37	1.81937168452765	1.7788180289167201	1.776338756811 04	1.78232832695241	1.76339568687525	1.9650025332705 99
BH	1.4583084516387002	1.47034471600059	1.40696526722184	1.50490032738466	1.49839375367712	1.44722993246437	1.4542064054455102	1.47001663687366	1.4069626299831302
BLA	1.6649240520799797	1.6262358918100202	1.6461511697676698	1.7435112816279699	1.69892742915382	1.716162608794 43	1.685657988143 42	1.627992940405 44	1.6454234745725298
BTS	0.9498168950560 56	1.02538485777859	0.9302225204336659	0.9918048416811929	1.00255277238989	0.991921742758033	0.9523986117918 57	1.01671203192444	0.9294448549456029
CBG	2.05526155761592	2.0249120495 52	2.14974369252512	2.06734369021687	2.04844173742213	2.03201215150419	2.05186296585039	2.02940629508056	2.133640746709
CENTEL	1.52183076580 13	1.60738846011994	1.4855055544017302	1.5597993855383299	1.54634002468199	1.529000629141 36	1.5213486412269699	1.57931732179869	1.485616255805 42
CK	1.3874323998658	1.4206991523643	1.3349510948898 2	1.41216199310396	1.41207386095284	1.347697006473 53	1.380939530003 03	1.4209413229063199	1.3348611292506
CPALL	1.3658355092074 6	1.3563450386086502	1.327509866702 01	1.35626629492301	1.36120539805048	1.343035729239 21	1.369112693318 24	1.3559808257652	1.3268451492212099
CPF	2.025827079911 31	1.9960054705488 5	2.0000373028666	2.0660409661630097	2.03604692782481	1.999889408820 41	2.034010231624 19	1.9960480035852002	2.000066636391 97
CPN	1.2869001205938 2	1.3065200148010 2	1.2494252659768499	1.2204528442907001	1.216602887214 8	1.219074340137329 8	1.279972968891 91	1.305875311554588	1.2499710507121902
DELTA	1.8397967434283302	1.88723557468131	1.7200247349058801	1.75717506370041	1.82565804441949	1.698999805376 74	1.836682314331539 8	1.8625463413272099	1.720438520948 6
DTAC	3.625037132400662	3.2595165148823	3.1410587452817302	3.2058994090087403	3.25071182944022	3.165511554710 73	3.626812934060 96	3.2612451536462297	3.1401226634010597
EGCO	0.950786207572934	0.936106899543145	0.9344610189959 92	0.9232990186086131	0.917919853972205	0.920449820610919 1	0.946405914941522	0.9340089290323 56	0.9342753844036531
GLOW	1.3177883153543002	1.281870761471 38	1.2535861438653 8	1.2842402938255	1.2458622091571199	1.271154309115 35	1.318322287647 74	1.2820123832298 4	1.2540900190917499
GPSC	1.480682644321 01	1.4558006574148301	1.591838674398 92	1.80635948841096	1.72455506098047 98	1.725630333731 89	1.480176933273 41	1.4999113588614 4	1.5692135365207902
HMPRO	1.5505642233246801	1.536670871166 39	1.5040978890639 7	1.51770726594401	1.48752875352416	1.474518557839 05	1.543807087544 9	1.5396603088284 8	1.5051752175218198
INTUCH	1.6302380156749798	1.6205124876445	1.619182461115 01	1.615855432124 89	1.61137683931377	1.602306296240 67	1.633111919677 6802	1.621400166500 67	1.6188295079244501
IRPC	1.4536389065183701	1.3714662215520 5	1.285822505465 33	1.4262513409742 2	1.38730477819187	1.302310889537 01	1.437879750128779 9	1.368395943934 52	1.285581965498 69
IVL	2.319475196742 01	2.2492137672782	2.0329866369678 9	2.33317498558391	2.23308134518023	2.060367226388 31	2.311325812658 31	2.2395763774915 1	2.0341729629563 6
KBANK	1.633413099924 2	1.5497467286567699	1.4154450262423501	1.4534574311751501	1.43816732441947	1.439796376859 38	1.634392204740 5899	1.542002097624 72	1.4146317263489 6
KCE	1.5724075035465201	1.560612634824 16	1.561387635103 65	1.4809094203859299	1.480410062597 6	1.454560582719 25	1.583066764614 4998	1.5555438606537302	1.5596602160513 3
KTB	1.212233080473 2	1.2693522100950 6	1.160739024571 7	1.2408682482593 99	1.21784679028785	1.168575796948 823	1.212371865530 79	1.2597911392353 9	1.1606345921032 9
LH	1.283054433743511	1.3289392027291 8	1.22774878574448	1.22397916855412	1.25131129283206	1.202104354815 6	1.277932115802 24	1.337827339324 18	1.2276581817379 9
MINT	1.79993122906238	1.7629696034467	1.6998371832470198	1.72187022435855 97	1.68680250138297	1.667009537877 46	1.812419496169270 1	1.7930608615039498	1.7003770079289 4
MTLS	1.0410334017796599	0.9378635583296839	0.9926110831398 86	1.0456886924575 4	0.9861814100695 56	0.984217042355584 9	1.085673818269 93	0.9957755906636 39	0.9932074357861 8
PS	1.3148015664336 2	1.3517529720479 2	1.2382237480583 3	1.2941370970576 4	1.27002327410017	1.221802557947 9401	1.313138982534 16	1.3694346389165	1.2385232855117 4
PTT	1.949748322525 59	1.9199179010771 9	1.8807960276392501	2.0041017873100 9	1.94953830841594	1.827229356216 54	1.948164675142079 8	1.9262219130146 4	1.8809091583930 2
PTTEP	2.2640058514158 3	2.3033929232415 86	2.2668841193055 7	2.2612904621544 7	2.26986431160804	2.190596859560 88	2.252501864361 76	2.2943829499114 7	2.2668756661699 3
PTTGC	1.825809737820 04	1.8074756845099 2	1.7678260685223899	1.8887427493718	1.86145343266679	1.812890398199009 9	1.841099933926209 7	1.8115110279440498	1.7669357000813 6
ROBINS	1.6879388803374098	1.6413183482249 2	1.5501675081540 1	1.6883390720136 4	1.64093116672809	1.506122072187899 8	1.699337826728 61	1.6399417892728801	1.5489085789518
SAWAD	1.4605283417816302	1.4410457752132 9	1.395399969545	1.464781649454 05	1.40940649467542 99	1.375645631991 79	1.457661579470 86	1.4461590173051901	1.3954085524212199
SCB	1.4988238508089 8	1.4982748787279	1.4418392028991 4	1.5141854143999 1	1.49345218142688	1.469228334870 02	1.496140789136 5	1.5114509943836 3	1.4418534597019
SCC	1.172964208814 36	1.1715183340464	1.0932473069006399	1.1451096132362 6	1.17559283085633 02	1.102248846192 92	1.169534287259 28	1.1608543405607	1.0932359857137
TASCO	2.533110817022 99	2.4835582273082	2.511298868451 08	2.5763112345464 2	2.54015261248653	2.496766407132 64	2.551108551696 37	2.4731813342611 1	2.5080815482133 6
TCAP	1.2238104310730 2	1.1889565857278	1.1947472408339 9	1.1939782961757 6	1.17385590566201	1.133377456823 48	1.197838524115 28	1.1872082535198	1.1944603806186
TMB	1.4125276712581 6	1.5040461495669 7	1.3815158425386 8	1.4295554174978 7	1.46295749209265	1.439278073114 7101	1.427028983928 46	1.5212334330143 63	1.3814195700592 2
TOP	1.5789033249769 2	1.5281063184134 2	1.5374807752102 3	1.5159705876496 2	1.51239878379529 9	1.532848262184 91	1.571395073859 84	1.5316391067494601	1.5413689476546 2
TPIPL	1.974568620400 14	1.9409820631039 1	1.584628678479 92	2.1064317347434 6	2.04383645445171	1.631537634325 48	1.982016481470829 9	1.8972497557348402	1.5841085139790898
TRUE	2.1184793858923	2.1486448822580 9	2.0176298881773	2.2171333279870 7	2.23566142982923	2.146270762706 89	2.162542164650 42	2.1442572231762	2.0114175848196 9
TTW	0.858213136103549	0.888619007002939	0.8418336898231811	0.8644733180264 88	0.84820618792421 7	0.833478863673843	0.851626345220675	0.8831831530250 76	0.8392372265116981
TU	1.353208357393 18	1.3885988368634 8	1.3136416756108	1.4162658691714 2	1.41267316386596	1.335453013969 76	1.350506725439 54	1.3958491297808 3	1.3136790214172
WHA	1.769080871302 68	1.8032633072652597	1.531874637423 34	1.7022495803214102	1.69376426087921 99	1.578794032212 22	1.688706606138 85	1.7635300489706698	1.5315243833125098

Table 27 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.775016950558014	1.761280130751350 1	1.763659763220 22	1.770201102725 83	1.765169479399 78	1.763013151370 57	1.762280069954 4399	1.781777297700 3	1.736290705980 95
AOT	1.162199123645 36	1.130764117696 23	1.152754832026 59	1.070022611026 26	1.06191599957 769	1.058547174769 94	1.168894378242 47	1.131973421367	1.152288566601 639
BA	0.931351757480153	0.958928761409041 9	0.893824027259099	1.138885116234 23	1.026875221430 06	0.963146270944 39	0.938146083297 136	0.955029348845576	0.893398893105822 1
BANPU	2.148565294398 74	2.109126728521 59	2.039870841938 4	2.088816278582 37	2.093665616944 8897	2.180082034885 52	2.137668079064 09	2.096803963933 18	2.039831548468 43
BBL	1.222733451237 92	1.217516067754	1.226220352308 2	1.268668922101 79	1.268584807186 3899	1.245352336155 306	1.227536650244 82	1.220886828930 47	1.225916342834 07
BCP	1.324691789944 58	1.353103016844 0201	1.335400518639 07	1.416902008556 89	1.395570032060 37	1.350755031518 11	1.330397621425 82	1.354603868813 49	1.335986870721 36
BDMS	0.993642599111 15	1.027140059889 8	1.033964060302 78	0.931905480220 171	0.901485468076037 9	0.954368675564 9681	0.982109359611 647	0.995146212622 92	1.030383089593 05
BEC	1.797869092630029 8	1.759820434176739 8	1.840122666480 1	1.844611032878 7	1.772276402718 79	1.692975050807 01	1.763205140915 93	1.759197365300 98	1.841217484449 81
BH	1.539089850446 76	1.550756024639 05	1.422630107027 73	1.555855199591 16	1.567500288644 8401	1.449092357658 6099	1.543348498753 98	1.554262533465 75	1.423251462912 66
BLA	1.674818973122 08	1.624985701905960 1	1.646218771800 4802	1.741350016717 64	1.698679583910 81	1.716160665206 51	1.675250789142 3	1.630482492413 95	1.645053291610 31
BTS	0.956338951422126	1.009576037689 11	0.931363855780 62	0.958256294607 774	0.993627054478 8991	0.979474652881 43	0.954229289819 504	1.009635526122 74	0.931098518012 664
CBG	2.057826373474 04	2.076114588054 61	2.149743692525 12	2.062581507262 83	2.049458971522 02	2.032012151504 19	2.062116663760 583	2.062742243997 47	2.133640746709 91
CENTEL	1.525076188368 16	1.565626911959 776	1.488504028026 65	1.551169456754 23	1.551603873672 46	1.536107990034 55	1.556295559757	1.601036761179 76	1.485639666264 793
CK	1.393486419036 4	1.404152767706 56	1.329037224144 57	1.415563529661 03	1.391752573976 1598	1.397295282349 38	1.385152975184 73	1.402343305074 29	1.327437940719 03
CPALL	1.369607736194 68	1.367204002377 42	1.327509866702 01	1.350068322035 54	1.346628756208 6798	1.343035729239 21	1.366119458234 26	1.361280819814 41	1.326845149221 2099
CPF	2.016249325968 1598	1.994799665171 1601	2.003454866193 6697	2.029701691199 4903	2.019434164414 19	1.980923607324 9602	2.017892615469 5	2.003340769280 103	2.003456992618 7
CPN	1.277586285777 6101	1.309425191416 4599	1.249425265976 8499	1.222805738910 1101	1.211550576212 7	1.219074340137 3298	1.274223817716 71	1.309332045266 12	1.249971050721 1902
DELTA	1.897176701307 9899	1.911056026939 06	1.720024734905 8801	1.803195745860 69	1.827797679402 86	1.698999805376 74	1.843604746614 2198	1.900878649525 09	1.720438520948 6
DTAC	3.157681256440 29	3.108864185734 46	3.097885715506 24	3.230045400016 7	3.247983034919 796	3.076589487612 36	3.155547795244 8703	3.106750461649 7	3.097545142750 57
EGCO	0.945757318522811	0.934484299134697 1	0.933522456241137	0.925573401053 51	0.917540690258 762	0.920261247409 95	0.945205214606 8699	0.938047574926 286	0.933319448125 2429
GLOW	1.319694126287 66	1.279290474862 53	1.253199204550 53	1.274397549216 87	1.243579777442 24	1.271057373660 97	1.328470413904 7	1.279960385170 15	1.253052149588 901
GPSC	1.526362121188 94	1.472087435267 502	1.591838674398 92	1.770817335055 12	1.737155474687 55	1.725630333731 89	1.517173357518 17	1.462054493488 51	1.569213536207902
HMPRO	1.553160669758 07	1.537875748730 1	1.505288046409	1.522326923950 27	1.487628160072 23	1.475339114889 47	1.541783786675 68	1.537371864059	1.506363606587 08
INTUCH	1.633262313686 24	1.621192019702 6701	1.619182461115 01	1.618474358652 11	1.612604868688 42	1.602306296240 67	1.627729137636 66	1.620098139973 2601	1.618829507924 4501
IRPC	1.442668721000 54	1.354833709663 298	1.278083190769 66	1.529013857012 999	1.447758536736 1399	1.304760748174 15	1.421992645850 298	1.354484382284 57	1.277803934058 198
IVL	2.328495625302 29	2.253104566868 57	2.044197138281 0498	2.272581653951 97	2.228992125098 52	2.182357228087 13	2.316929620619 75	2.250297575274 2302	2.043753292878
KBANK	1.534914386850 59	1.479661485881 38	1.424608875551 57	1.495221405777 52	1.458160868362 71	1.466899501014 5	1.527815565012 59	1.493902469258 75	1.423916558167 0499
KCE	1.575651104395 07	1.540373895241 6999	1.550439665070 6198	1.548067707899 61	1.480998762935 33	1.454021367996 66	1.592523172493 12	1.604338215886 8298	1.549738939424 07
KTB	1.218675154878	1.270399391579 99	1.160771707746 51	1.253069995556 57	1.215462103820 3801	1.168575796948 23	1.223788071593 35	1.269489573509 44	1.160634592103 29
LH	1.297553092633 35	1.343161505137 7501	1.239405062322 68	1.285039947636 91	1.323368126938 24	1.240490112227 3901	1.315538362584 41	1.319184626237 44	1.239078678014 6599
MINT	1.857918812665 3001	1.790650741737 31	1.675118273047 19	1.726091721515 64	1.697697743756 18	1.672994093183 2302	1.871083390236 01	1.795204738462	1.677066169547 68
MTLS	1.091567293460 36	0.964408505302 7339	0.992611083139 88	1.038503987851 5	0.966323545916 273	0.984217042355 587	1.035653575194 3301	1.052301627237 11	0.993207435786 1771
PS	1.310578249378 3902	1.396606340955 98	1.235563965839 2101	1.308332990744 6	1.276124096606 9	1.267967864867 03	1.323530568690 71	1.396771291123	1.234953489868 67
PTT	1.927957512205 18	1.923176900814 401	1.883319981690 16	2.001989983423 68	1.950988723071 92	1.829195540441 38	1.916852563514 51	1.923555743494 29	1.883387106410 6202
PTTEP	2.204835627336 6503	2.223707522182 92	2.222788901306 3	2.241754073157 58	2.134566492876 18	2.179972220586 81	2.203298077113 7503	2.224502564125 3	2.222719887048 03
PTTGC	1.844699935241 16	1.806269937378 802	1.767826068522 38	1.893923750624 4402	1.868580678079 94	1.812890398199 0099	1.825860748491 553	1.811279988803 183	1.766935700081 36
ROBINS	1.687894434500 73	1.640469097064 601	1.550167508154 1	1.687122022059 1499	1.640635158114 78	1.506122072187 89	1.676346011006 32	1.642095886434 6797	1.548908558789 518
SAWAD	1.452899320073798	1.445351866349 026	1.395399096954 678	1.465288860344 71	1.413896777656 08	1.375692993131 57	1.464991284718 35	1.445608949176 64	1.395408552423
SCB	1.49738214897 4	1.519799049166 58	1.441839202899 13	1.514450262526	1.502793719158 66	1.469228334870 02	1.503998737916 8098	1.520250801230 83	1.441853453970 19
SCC	1.174815289831 94	1.169102719311 81	1.093043062613 77	1.167898350558 76	1.160395277226 1399	1.098684118045 9002	1.174662504824 57	1.172794004063 53	1.092852282337 37
TASCO	2.910793338331 17	2.709802766958 78	2.620921756391 72	2.620580071644 92	2.525384003890 08	2.496629844843 18	2.880745814154 73	2.716497315872 8003	2.609494461882 24
TCAP	1.197346998571 13	1.187349723373 37	1.194747240833 99	1.199260741991 32	1.171503512521 7	1.133377456823 48	1.195385726581 08	1.189043321600 13	1.194486325994 93
TMB	1.430484570329 52	1.514629657239 869	1.383003590055 61	1.441734309022 32	1.467435920966 0002	1.432028324269 13	1.416598135235 63	1.517297394579 04	1.383339673816 4998
TOP	1.574761747342 07	1.519301343199 74	1.537480775210 23	1.516524510273 15	1.511132411191 27	1.532848262184 91	1.575205157830 77	1.518233106494 76	1.541368947654 62
TPIPL	1.821441770020509 9	1.922712251944 5299	1.565136461727 7501	2.028263478555 37	2.069672854480 94	1.693197595129	1.855707533079 09	1.910084863893 35	1.564629421655 24
TRUE	2.086206033238 158	2.089011312797 6497	2.002491842306 81	2.140741857245 12	2.126177730902 35	2.010797030377	2.104596375393 72	2.090818290022	2.001266127542 21
TTW	0.879690776218392	0.899008491105 0239	0.831508656090 808	1.037528293948 3799	0.931319975250 2169	0.815238022873 0639	0.868411954524 9219	0.897284388216 501	0.830082130128 217
TU	1.350417177203 18	1.383772907735	1.313641675610 8	1.378604023035 67	1.387184155341 89	1.335453013969 75	1.345622122405 61	1.383121447418 79	1.313679021742 61
WHA	1.746802178456 95	1.781128940454 899	1.525934427758 08	1.669286257036 05	1.697460156565 87	1.586999461953 81	1.690461875831 38	1.710181820702 1497	1.526290484861 65

PredictPrice10 : RSI (Relative Strength Index)

Table 28 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	33.2093822826525	32.4490474731862	31.423471781959	32.9192158103 86	32.5793508819428	32.1485724790819	33.164070689973904	33.036657659085	31.59766388972 75
AOT	29.6498445247518	28.3028244562442 03	16.601740262570 7	30.1386268269822	29.4211254063183	17.2988646982378	30.039330809876503	28.4948151818 73	16.4973213068211
BA	1.20340387167895	0.9966179237021761	1.1017371454765401	1.0258927789044	1.0208221172611 2	0.990022127668848	1.12839786954144	0.995009263578 41	1.1021593935 4593
BANPU	45.298801683338006	36.22221071315 49	19.667871700337 3	44.778695937741	40.76255508399 55	23.195786594221097	44.20741231377 82	43.807256209418	19.8923149035 235
BBL	6.37432016801233	6.078045322471 86	2.561457728852 92	7.046252911784 31	6.3766159079383105	3.1647691786267003	6.007059370541739	5.095734517484 14	2.5615872860 9352
BCP	1.7214009845495302	1.797156765007 94	1.421890762400 61	2.062875058446 95	1.9569551324055	1.4225355790143799	1.69715487798462 02	1.762106922734 01	1.4078745956 2285
BDMS	15.8676706869795	15.5649847478458	6.527706578215 9	16.616722827364 9	14.942525451799499	6.84727863292 48	17.27769920300 99	15.021957041333 3	6.4827435495 81679
BEC	44.8721077283122	38.7331653343358	15.064738726808 2	40.117877641650 8	40.2790610792599	17.0521164546337	42.312003899387 6	42.026128233425 9	15.1712451480 229
BH	3.27969313167241	2.7280939276048 6	1.9193425387737 3	3.4077625575248 1	2.8862298730 5008	1.9349014823737898	3.500258021776 16	2.922486888042 8803	1.9020214623 997
BLA	21.1073982295475	17.5595301627093	7.536915335929301	19.835121675754 4	21.7162194261844	9.063590078854 34	18.843322564856 1	18.959527516661 502	7.4704899904 1367
BTS	1.01190243494197	1.0654381301459 1	0.961272939251145 1	1.0553570162253 4	1.0573421569176 2	1.0197540467141 502	1.01668274060077	1.049572298662 47	0.9614859344 707001
CBG	28.5772998531884	28.4573630926187	26.407548712357997	28.626949092209 9	28.400472862494603	26.259582776582803	28.526242919724698	28.590918657929198	26.1880607273004
CENTEL	8.22767343429607	8.1147112046457 9	3.8860589037981	8.2774878496209 2	8.9858196467334 8	4.6075422903990 6	8.726293967650 63	8.037223026140 1	3.8509614618762997
CK	4.042913313121718	5.0675650142581 2	2.3965147674709 2	4.5501656153341 7	4.3536386981054	2.4922170130133 8	4.229744710893 39	4.480481100667 6	2.3761508758976 3
CPALL	2.69413636457432	2.6913566732177396	1.6611042898245 4	2.7425810825054 2	2.5911262161601	1.7895738164284	2.636639404084 23	2.623756209233 43	1.6615077771749
CPF	6.0972736066244195	5.0734045376596	3.097344961070 33	5.1668595894573 2	6.5729083597624 19	3.0898661038940 2	5.502443451091 76	6.695336532587 83	3.1115391940091 7
CPN	10.7742512234679	10.501487158083	7.726585981473 7	10.939784586032 6	10.714721455125801	8.7394529964981 09	10.687987598431 3	10.861029316443 4	7.6139674295417095
DELTA	2.22959303119882	2.3996859093496	1.9563376633185299	2.2111687606635 4	2.3306523936340 91	1.969358104392 76	2.438729560161 84	2.338163380828 87	1.9554235852201
DTAC	137.335216683061	127.539436633618 99	67.099112787133 49	142.226089076819	131.911559692949	72.143549723488 29	136.278582859402	129.118700918117 98	67.362108365121 2
EGCO	9.028439786376 09	7.942109685122251	3.841353083192369 7	7.703901396675040 5	7.922325745436309	4.052998309940381	8.240592059522 52	8.034003988769 21	3.8305333379438298
GLOW	1.80285962605806	1.7545957939408	1.416473789143 47	1.9250816431942201	1.8413202247131 1	1.611151461331 52	1.726840864644 49	1.667099116417 26	1.4223215250908 3
GPSC	5.747657761658 74	5.5022375059510 6	2.160651635274420 3	7.158478658496 73	4.885085112875 44	2.626130705313 91	3.852404118024709 5	7.395403280884	2.3401278892623 5
HMPRO	9.655526592837 75	5.2203325197342 3	3.311581679546309 8	5.517302074814001	5.563965415936 19	3.416913057358104	5.383253426567 98	5.228665574490895	3.3111520240179 7
INTUCH	28.952353902341 5	28.380060535045 4	29.017846125156698	28.869037854940 8	28.789700465081204	29.840371299215 5	29.126410935276 7	28.360744339041	29.1906345941239
IRPC	11.343455200566 3	12.097295017127 8	5.581338074608 36	11.827578139609 3	11.894246977767 9	5.986904422516 85	12.067117687180 8	10.110325436289 2	5.6237933499 14
IVL	9.441011382906 96	8.9537207281059	5.496246111097 17	8.995337356782 57	8.795491498767 71	5.896283441382 96	9.012010190181 31	8.224290415240 33	5.4829100936904 7
KBANK	3.022208823061929 7	3.543507519987 04	2.186421263532	2.799695033123732	3.697932736112140 4	2.309681315372 8397	3.573819915252 86	4.274487723487 76	2.1869954469 3929
KCE	40.404230076637 1	39.121060409590 1	25.505247642046502	41.688303264356 3	39.757497026383994	26.861298108092	41.348340749134 5	37.619418483643 45	25.348204478 2229
KTB	1.896381252226 24	2.283319688822 518	1.268457499978 2	1.765027935352 29	2.987710690152850 3	1.341768237766480 1	1.842232043639 2798	3.756124519887 7	1.2824405409 7898
LH	1.3444774122551	1.3984808592325 3	1.3554346954499399	1.342932891077 44	1.350435597573 01	1.232372745939 2	1.308333724959 08	1.396593954947 52	1.3568053638 818
MINT	21.2944282894912	20.7934113764498	10.120381635103 99	21.109017614560 4	21.587172406180997	10.633360314332	20.652958948280 1	21.624517490872 1	10.0044195421 709
MTLS	3.5772479440947302	3.2571985586245598	2.3898693688778	3.7481881708938998	3.552407712459589 6	2.734348139112 75	3.536864291539 79	3.506862651132	2.3872783709 046
PS	1.284329393286 66	1.3546842566214	1.373191783678 97	1.3491150282998698	1.336263101387 66	1.421323021201719 8	1.347945514589 26	1.339900072361 76	1.3737364326 9326
PTT	6.469638359786 6	6.7517434185794105	3.116930581854 3	6.840079239072 88	7.241391965170 89	3.789145872268 95	8.201992427685 46	7.757585666114 05	3.1386979349246302
PTTEP	32.425249490261 4	43.649431855681 8	10.218894584776 5	30.615106702303702	35.582895895580805	10.434348147303101	32.974905537848 1	29.378617531794298	10.3009248297 423
PTTGC	4.849303528035 71	5.400895333268295	2.924881979706 7	5.352105802490 69	4.9871846302 2141	3.229243581525 02	4.873265714140 6	5.251551348501 96	2.9210420088716003
ROBINS	6.858233222996871	6.42693633753	3.084319512962 04	6.132587417615650 5	6.684802294824 86	3.503093955913 73	6.688556790149 169	6.226948896703 86	3.0828165868 1498
SAWAD	1.7783326061179199	1.8116865848538701	1.6361319608297 8	1.8890403083561098	1.8057119976273102	1.613677813998 53	1.869195670288 3001	2.227391593593 66	1.6331119400 4514
SCB	10.14001132274 69	13.4199682369834	4.775933935429 72	12.4406502785655	11.387015570300 1	5.442004807964991	10.898453258391	11.320415570816799	4.7720074257355 8
SCC	1.2607584740029198	1.2486896393837	1.152212309887 39	1.2874823821750399	1.3056155164524 2	1.140401474040 02	1.191675672589401	1.266416359826 22	1.1523791423 418
TASCO	25.163951280059 7	25.350097136698203	9.392127941567 23	25.150972821981 8	30.499411170833 6	9.695754284298 4	25.821487959164 2	25.655502087360798	9.71003153665
TCAP	3.511798597589 32	3.6327239944208403	1.8405143275418 8	3.614076827958 5	4.127797345879 42	2.212786604521890 2	3.761176676828 0903	3.251116596982 43	1.8397907300 432
TMB	1.559030400712486	1.5483971019597 2	1.511962841005 8	1.525747866278 05	1.538015000332296	1.522021739766 41	1.597361143039 29	1.543693889099 489	1.5108711236 0448
TOP	20.782202285139	20.369960619239	14.219525839988002	20.6898180993156	20.590479136018 9	15.070526784904 01	20.586572070860 02	20.523464649297	14.0449738041 881
TPIPL	2.465108405228 96	2.2273887891266	2.333046765022	2.370572514115 27	2.190049338854 8803	2.271908304876369 7	2.356432563911 9	2.220861960695 62	2.3393124166756802
TRUE	5.279979150398 06	9.103199734924 22	2.719210719671030 2	3.757240060573 91	3.931110543178 53	2.726439091966 2	5.276343655503 3	3.069822681909 4	2.7230800017056
TTW	1.437238919200836	1.078471298151 57	1.040065297373 178	1.221636828982 63	1.092596621675 2	1.027291555666 956	1.138720962173 65	1.038387833061 95	1.0397550629116599
TU	1.509684130712 94	1.530375369508 14	1.364739503344 361	1.450175466159 54	1.547174321506 74	1.354392716850 99	1.456835951831 56	1.507155413133 28	1.3647594957 9049
WHA	1.7914323650604198	1.8421139198938101	1.656642598563 29	2.206431124436549 8	1.859444926078050 1	1.709823451659 68	1.808218314966 18	1.816841876216 1	1.6514477094 0641

Table 29 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.887701128696 73	1.9019001448936	1.7795786518190402	1.8550104815327 1	1.8346977648323899	1.772011221071 04	1.885867880 37	1.908452471272 78	1.7778126061304 8
AOT	1.071187737636 3	1.0438499240043 2	1.0333645680008199	1.0523447666871502	1.040764003853 09	1.044393804491 73	1.065411242189 54	1.044806941909 64	1.033422006661
BA	0.9561563755360389	0.884725231703145	0.9753726839207749	0.9776401941775179	0.893158490810733	0.901725288314 87	0.967541965420743 1	0.884183103070959	0.9749032368935 85
BANPU	2.125691849866 19	2.0956457904135 3	2.01082077507965	2.113003605190 22	2.1004399955557402	2.004879206170789 6	2.083289006712 15	2.115332929208 11	2.0105482719147 8
BBL	1.214257121119 44	1.2121242094448 8	1.2178257947834699	1.2375344438933 3	1.2334389583763898	1.227292276247 51	1.217745917077 89	1.208033531307 37	1.2172301400773 8
BCP	1.378103383573 76	1.3723070375743 5	1.336572231952 19	1.3834156988030402	1.3833445414832402	1.307710707043 9	1.388449732075 56	1.378000322007 82	1.3366346183197 4
BDMS	0.899352178251 65	0.9277807485685 41	0.882516775328209	0.9183667480615859	0.9236044572859019	0.903537867966866	0.904138577231 23	0.928727732851 0609	0.8823615123288 02
BEC	1.687706781583 38	1.762392345433 69	1.6562550793545099	1.771268349119 16	1.870930701071 77	1.685954517654 78	1.729888953515201	1.749851344887 71	1.6569489825642
BH	1.465870832565 03	1.4302360438671 7	1.395510771431 81	1.486208496616 4	1.477926586254 01	1.481013325747 3	1.441907561934 4	1.428574832825 7801	1.3954588390905 5
BLA	1.633192207565 55	1.6792697701093	1.6429817733787901	1.684919552629 53	1.717357275846402	1.742696705935 53	1.636523049930 92	1.675874538827 9501	1.6432347222889 4
BTS	0.951997470561317 9	0.975750153389516 1	0.92511405011334	0.970163214049917	0.962958253780851	0.976360178607854	0.952025283281876	0.969363915288733	0.9257640812611949
CBG	2.213805426807 22	2.159038176462 4	2.053633230784 29	2.173987750 3108	2.078215614848 63	2.110981744493197	2.191855543333 25	2.181777230299 74	2.052878169263 5
CENTEL	1.579507239653 51	1.646463691790 2	1.5069843560363698	1.5882669160907699	1.600890477070349 8	1.544226581965 02	1.566087348468 7001	1.647672989197 02	1.5067747394674 4
CK	1.361853443456 9	1.4467812907303699	1.347730688516 18	1.3691873575878402	1.470452607534 1	1.351187950606 41	1.373476425082 698	1.448533160323 04	1.3476924602685398
CPALL	1.380930587081 24	1.4124700246219999	1.3173099075438999	1.337823013093 1	1.3603214080041	1.318153389950 03	1.386863971982 1	1.406442401759130 2	1.3172869682239499
CPF	2.083958310854 54	2.0856013846327097	1.9941216377069901	2.044140770774 49	2.035121634245 49	1.988226215563109 9	2.081047316631 77	2.088490642391 64	1.9937969072476198
CPN	1.244126193712 87	1.2438226489825102	1.240873245171 8	1.2306771768238001	1.233367778570 75	1.202965899915 97	1.241574363732 1	1.244999107076 64	1.2408764213570 3
DELTA	1.852233565894189 7	1.8649629547504603	1.7045123673361602	1.863485990620 274	1.8631826412646502	1.721783593067 22	1.841159674003 82	1.886052989093 746	1.7046267918889701
DTAC	3.058558606828 87	3.2250910406026	3.048818927260 06	3.27141722970 76	3.251582540324510 4	3.110587255571 08	3.058177764411 5	3.224098970635 8603	3.0484637783 713
EGCO	0.926657803470941	0.9316884017502189	0.9137388988993571	0.922685498556839	0.909853991664505 1	0.909678297155961	0.920811592393386	0.932018728801441	0.9123415545746 04
GLOW	1.283597192407399	1.2927807760937 3	1.2491510729133701	1.373005827616 51	1.3281355548977 1	1.331288150853 69	1.258916984477 33	1.313239094225 3	1.2495971792763 4
GPSC	1.55855626655 58	1.6146083410200198	1.7010264474718502	1.900423228853 698	1.822442401205219 8	1.723423632157 75	1.567845444494 088	1.658885921472 22	1.7013713150642902
HMPRO	1.512388629745 69	1.4913583112606 2	1.463232067944 26	1.535664736658 2399	1.4996225270748 3	1.473172437368 86	1.504378357804 13	1.483445883889 485	1.4632540412638 8
INTUCH	1.631425483360 14	1.6327562572743999	1.613853505646 76	1.683822193775 06	1.664629488693 51	1.642215267829659 8	1.630438833920 608	1.632114690496 38	1.6126304047612
IRPC	1.354433001197 7799	1.3518165328205	1.294636223494 7	1.342128219597 6	1.353692797434 09	1.295399621574 09	1.351258269292 22	1.354110364611 048	1.2946784157281902
IVL	2.167729750962 1702	2.235424952862 65	2.107716860242 47	2.171057440557 63	2.199825642276 04	2.129647712934189 8	2.152073060402 42	2.249862882642 74	2.1082338484051 9
KBANK	1.378999022468 6	1.4163423969164999	1.395798360134	1.4245419322180999	1.4200721929471198	1.440793045495 35	1.384454132466	1.409428353806 41	1.3956057333479
KCE	1.503234700969 24	1.4886126621587499	1.46009511738 24	1.502005282953 1	1.4742818067432901	1.454046973204 61	1.503480955274 6	1.501067907886 5399	1.4592469224497802
KTB	1.252415639013 5601	1.2609834810141 6	1.150583095156 66	1.255756775298 47	1.266993526866 3	1.188013484093 03	1.203081095008 58	1.257593429927 92	1.1505918992985 4
LH	1.333643055197 64	1.3036293975181 24	1.228545015401 04	1.2317570543202099	1.261218336811 16	1.199849094204698	1.299997868717 23	1.308263119783 19	1.2284374030882 599
MINT	1.628910891981398	1.6233684083522 4	1.581636552552 82	1.654362639417 19	1.644272913426 66	1.594838270610 44	1.631230167970199	1.623645512125601	1.5811798354422 02
MTLS	0.976725061425 35	1.0556331544076 6	0.9643327628713271	1.178109225921 47	1.116666010693 91	0.991138705858793	0.973943039827981 9	1.046700286345540 1	0.9651324816025 31
PS	1.255249071859 96	1.2935811866948 7	1.233014940125 1	1.263383390492 1	1.317174456789 34	1.213568420255 13	1.248168721332 32	1.322195139146	1.2325909093691 65
PTT	1.848670747117 76	1.8291600063197 8	1.848897057009 4	1.823026960467 66	1.812582855294080 1	1.805571038088929 8	1.866022835205 96	1.829149830225 89	1.8490146489157901
PTTEP	2.348233235323898	2.437279224680 99	2.310971133575 03	2.232447483960 56	2.336539200358439 8	2.209263792236 33	2.385754368936 96	2.464228149794 54	2.3100483283704 3
PTTGC	1.799362232346 5	1.8295062680648	1.761853743223 81	1.839359943203559 9	1.840813211358539 8	1.795854421109 57	1.802923506518 7	1.804896931743602	1.7616632655026 4
ROBINS	1.498064497562 97	1.5432107523328 8	1.488590546846 85	1.520605654215 41	1.512260986862 97	1.468644400251	1.521637246036 17	1.541097319894 45	1.4884272864521 3
SAWAD	1.530894596707 73	1.5530614099188	1.427071154747	1.493937332363 64	1.490533236364	1.406533295213 11	1.533367853603 67	1.553187215978 4001	1.4252261302935 3
SCB	1.449520817052 41	1.4556145048092601	1.433323183740 88	1.530084349442 01	1.527246350312 81	1.464707668102 79	1.452138644289 7	1.452531361083399 9	1.4332389821099 8
SCC	1.119685755718 4	1.1036488433776 3	1.083122353218 11	1.116396075662 92	1.096263343251 732	1.088827695772 94	1.119304750768 89	1.103144653520 71	1.0832913069092298
TASCO	2.448300913602 93	2.421745470225 69	2.345782318319 97	2.600318924184803	2.575017775235 14	2.475048135715 81	2.454183232360 134	2.446735663507 18	2.3457145804991 7
TCAP	1.140136624957 76	1.1627597560457 4	1.113680661845000 1	1.182924857457 6	1.176066802900 58	1.125836822667 3	1.140690442206 82	1.159458287699 601	1.1137237485317 4
TMB	1.487114622910 35	1.5217564360518099	1.398942650537 09	1.497695607154 54	1.516079480133098	1.514337470125 7	1.481973756286 54	1.534315480764 94	1.3990657076435 5
TOP	1.521515281333 88	1.5140924242779 8	1.508092255088 42	1.548319794865 68	1.554966310008 32	1.506424666921 59	1.511769177903 91	1.525520065513 21	1.5082625878218 6
TPIPL	1.644439921156 4	1.8611364979788 2	1.632344749638 25	2.019146989593 02	2.020330584233 14	1.776876788256 37	1.645392116940 91	1.786710256853259 8	1.6324900563162
TRUE	2.095335427911 44	2.1077464483981 8	2.091026085027 95	2.112429767879 88	2.146906463406 63	2.018852181329	2.065075050421 65	2.116028250514 21	2.0897467830050203
TTW	0.842262737512581 1	0.847736285797219	0.811995172860773	0.869480606600969	0.847885989381265	0.832214768170497 9	0.851974950001952 1	0.846499408808056	0.8122977333257011
TU	1.331590465817 94	1.3869945688415 5	1.318661255337229 9	1.338353211542150 1	1.347564046963 39	1.329114406041 91	1.335399265507 32	1.378066910767 3	1.3185036773125
WHA	1.655800468395059 8	1.732447319972 56	1.520585296898 01	1.805499635830399	1.769147432939 58	1.578827852446 4	1.658402361298 26	1.701962649782 83	1.5212380249077 6

Table 30 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.807551428795041	1.8103528926979602	1.775790144599541	1.7892573220784702	1.7868008791580001	1.7790484640114699	1.8040323821023	1.8090241837810	1.77294267956353
AOT	1.08827607096379	1.0941925607838001	1.031404448123941	1.10769408543996	1.08357817038255	1.0373940028414999	1.073255100938452	1.0960472863155502	1.031819089677 31
BA	0.9204153278441939	0.9171326004387951	0.9487940043410401	0.983563410749 47	0.946734011929661	0.9108542506327 02	0.925488419022957	0.9158318855284583	0.9491100407733942
BANPU	2.082858017850055	2.0903654296511602	2.014978961571 14	2.136290547185 51	2.107954038063 61	2.022355416496 54	2.105645500929153	2.107348866514404	2.014993676229275
BBL	1.212480377410 03	1.1982113392331402	1.214850656487 12	1.226316733239 68	1.222985522387 52	1.233506972747 5299	1.216203543643 2101	1.194308190454 83	1.213937931000 68
BCP	1.368614577259 14	1.385697697951 46	1.336572231952 19	1.386250384048 92	1.384895263858 09	1.307710707043 9	1.381432451080 6	1.381981555702 069	1.336634616319 74
BDMS	0.8970439000460261	0.928124380030544	0.882255019623264 11	0.921425820519727	0.924845215775369	0.903445118449891	0.899300362483464	0.927635769292652	0.882315758668706
BEC	1.6535087791193799	1.74627416273896	1.651187527756 58	1.790594715268 93	1.834091388774999	1.679725449283 11	1.656260621788 09	1.730778696867 14	1.651747864320 88
BH	1.461919954511 64	1.437989087105170 1	1.40456557948 75	1.477355590418 15	1.488171579272299	1.530602749799 8098	1.453856317389 48	1.438496487063 09	1.404151788222 751
BLA	1.633676922300947	1.684299704442539 8	1.642908039720 37	1.712008066331 56	1.717156619533 03	1.742669675935 54	1.641487414088 6	1.682633811436249	1.643324722289 5402
BTS	0.9307424548121781	0.9383168762819191	0.913839835702625	0.938381058397496	0.939128143751550 1	0.977350246347 34	0.935435402483849	0.941896152234786	0.914024922802204
CBG	2.220639928661 49	2.152892900848 88	2.053633230784 29	2.209596006291 23	2.102797112336 85	2.110981744449 3197	2.210762766253 0502	2.159888241958 31	2.052878716926 35
CENTEL	1.575741844350 01	1.655021416495 47	1.513771571374 49	1.593432554409620 2	1.602756229797 58	1.545973100286 31	1.573368326705 41	1.655303398295249 9	1.513645021391 6801
CK	1.490668097105 74	1.540633410119 85	1.382298613248 5802	1.429719347523 97	1.483644387433 33	1.343846496338 5	1.448524450295 02	1.501850405746 1	1.380917608038 17
CPALL	1.359494563004 99	1.398279717272 71	1.317309907543 8999	1.351143301479 05	1.355073094902 81	1.318153389950 02	1.360865454868 42	1.408226539181 59	1.317286968223 9499
CPF	2.028815636443 35	1.996431988332 7701	1.995817256425 1202	2.026805838055 18	1.994795090851 8302	1.999364286757 95	2.037531863908 39	1.998883222670 12	1.995904600511 23
CPN	1.243147694562 89	1.246259520254 13	1.240873245171 81	1.231307439811 96	1.227994282128 23	1.202965899915 97	1.240480323038 21	1.245330019297 83	1.240876421357 03
DELTA	1.818635609076 8	1.878418622664 3103	1.704512367336 1602	1.850971027099 35	1.878602147899 0298	1.721783593067 22	1.829165734277 01	1.870545736435 46	1.704635572673 37
DTAC	3.140666181825 5	3.145562684988 78	3.058614740996 03	3.186874229217 5096	3.194574167611 2096	3.098006188916 47	3.129953890145	3.139931314387 34	3.058002895899 28
EGCO	0.9190025209222069	0.9290133561098981	0.914252320651871	0.913684356703040 9	0.909236212275292	0.909763601251874	0.922365483797 6941	0.928941354866 598	0.912903930236 24
GLOW	1.261349921495 5601	1.293753217458 82	1.249092766865 84	1.375311386345 29	1.328297687848 7098	1.330215751471 39	1.249783844661 73	1.293602101634 1301	1.249351263401 9999
GPSC	1.613150051610 8601	1.632242829507	1.701026447471 8502	1.885143642908 4	1.822675345530 22	1.723423632157 74	1.806226534999 51	1.604659360025 49	1.701371315064 3002
HMPRO	1.522150599101 33	1.516161436697 57	1.465101895999 86	1.542566698019 7599	1.509455847079 33	1.474105743353 48	1.503607101494 86	1.506222502928 69	1.465113201258 97
INTUCH	1.631102917241 56	1.632919473839 0298	1.613853505640 76	1.682612518119 0699	1.665660113087 7298	1.642215267829 6598	1.633751505386 07	1.630650070111 03	1.612630404761 2
IRPC	1.379236977561 2001	1.382937028326 76	1.299278983394 3599	1.329495456828 2	1.358512117321 4099	1.286840916891 51	1.369941902182 93	1.385087587775 6998	1.299339679755 33
IVL	2.188039022097 77	2.204177975276 1	2.076409516550 95	2.272890406445 57	2.243005877497 598	2.114729277858 21	2.187616181973 31	2.209295923998 5	2.076505599265 64
KBANK	1.402266506837 65	1.409651070081 47	1.420840988021 6	1.427628532477 226	1.413618600234 05	1.437719971591 52	1.409176876728 44	1.406786103795 23	1.420704514269 6302
KCE	1.488286806816 63	1.504867865598 27	1.459787203909 26	1.499095973204 71	1.465532528284 9498	1.459009643512 0299	1.488447600043 23	1.500278212516 801	1.459826248833 82
KTB	1.208425227340 6	1.281468943892 43	1.150537596545 7099	1.256455269623 15	1.267872495216 43	1.188013484093 73	1.207856894266 23	1.291707892590 6	1.150591899299 24
LH	1.278687069721 02	1.278045414530 31	1.227445484013 06	1.270435747416 6201	1.267009975369499 9	1.223088016571 86	1.272443021899 48	1.281171487897 83	1.226986223535 14999
MINT	1.627779764590 7702	1.631316928432 84	1.585294765844 2602	1.623463969570 05	1.623408945922 81	1.601209407573 36	1.628740626690 04	1.631934823758 42	1.582770081054 3701
MTLS	0.9848436792032 26	1.084333468419 5	0.964278409561 0479	1.097206560815 43	1.109348761172 19	0.991138705858 798	0.985765375689 0661	1.028105798306 97	0.965132481602 5391
PS	1.308730922290 34	1.372214041622 926	1.263229809043 76	1.315555936426 16	1.380989932200 28	1.227111735360 46	1.303460777019 15	1.390647335552 3399	1.263153128164 29
PTT	1.855939628113 52	1.830950736668 88	1.849691893331 7401	1.823617504810 3103	1.810018097233 5098	1.803544631389 72	1.855021081523 99	1.830108482355 3402	1.849736108436 15
PTTEP	2.269651184675 7796	2.310580570450 43	2.245408096933 2	2.170874886182 99	2.186678856536 52	2.218283331508 2	2.273715162587 2	2.307753388046 1	2.244513188402 91
PTTGC	1.805986389445 07	1.815548413562 3199	1.761853743223 81	1.832979397306 08	1.840808729705 7198	1.795854421109 56	1.811095939168 3001	1.837977325040 83	1.761663265502 64
ROBINS	1.521904633046 36	1.539906634878 6201	1.488590544684 85	1.509551320347 6498	1.512275876042 83	1.468644440025 1	1.503317048087 12	1.542996886237 29	1.488427286452 13
SAWAD	1.527468305973 0298	1.553129546053	1.427071154747	1.460760297319 6698	1.489865247326 07	1.406533295213 11	1.552886804227 71	1.546269604561 7	1.425226130293 52
SCB	1.452160597666 01	1.451117082549 8698	1.433323183740 88	1.528996141627 1198	1.527471891270 57	1.464707661802 79	1.449564518189 7399	1.457638307002 4699	1.433238982109 98
SCC	1.143250681298 69	1.113303248737 35	1.082069093280 8101	1.122857909335 83	1.098468849245 08	1.096424415904 1001	1.137217783813 702	1.115529559485 51	1.082844194187 54
TASCO	2.361278617772 3	2.423621373798 02	2.331585062346 64	2.656245454357 27	2.620266553189 7	2.629900371387 0497	2.358874952592 15	2.424768616566 142	2.330730520961 6
TCAP	1.141874578661 58	1.156927613276 94	1.113680661844 99	1.187284096989 13	1.175803477492 96	1.125836822667 73	1.140665299287 96	1.160690555331 47	1.113723748531 74
TMB	1.478483332533 61	1.541208856038 4	1.398604628888 28	1.513402929932 97	1.509643954343 73	1.516327481785 85	1.483854823207 03	1.549454079371 98	1.398515628376 1502
TOP	1.523738330377 36	1.514069483753 2	1.508092255088 42	1.534371437138 99	1.566528470205 17	1.506424666921 59	1.519162595823 74	1.514136014676 66	1.508262587281 86
TPIPL	1.763336078170 34	1.940045919764 79	1.646759423110 98	2.001679565331 31	2.099357638438 94	1.759903338554 98	1.832830451727 89	1.913682698120 36	1.646315686242 9398
TRUE	2.013574874878 1697	2.088974088457 2396	2.040295390578 4402	2.032268657211 58	2.070361131835 63	1.986770601341 7599	2.021430310624 73	2.073302163011 03	2.039709747748 55
TTW	0.816567690618 466	0.846469370225 1931	0.808455847084 303	0.870763654820 975	0.858449991710 6939	0.808461849546 851	0.816699947162 468	0.846859012999 126	0.808798681492 141
TU	1.332006200275 54	1.377612563645 39	1.318633653285 67	1.342879174428 32	1.359211355147 09	1.329114406041 92	1.328649576320 64	1.378874701152 73	1.318503677312 15
WHA	1.649188037289 51	1.683590250407 54	1.527525897999 67	1.790324099280 3702	1.780303231976 0599	1.597829400946 74	1.653356260458 92	1.732827778380 92	1.526960460572 2

PredictPrice11 : ADX (Average Directional Index), DIP (Directional Index Plus), and DIM (Directional Index Minus)

Table 31 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	34.44912104184 42	27.41662663739 16	36.18812478982 8905	33.57002746019 73	26.53525174484 1	36.13691333849 009	34.44456573793 77	27.02321170125 9804	36.19007075425 28
AOT	25.25537858527 0698	18.88998307161 68	21.15540612480 34	23.16434167612 22	16.95000021267 95	21.36893187094 43	23.09828741772 6198	17.79248605314 73	21.05329848561
BA	1.13947130799 09	1.05124142414 737	1.00727326598 86598	1.38244218401 7301	1.33049008035 386	1.30194950590 41901	1.28194574799 428	1.04923387066 14601	1.01316135083 509
BANPU	37.84160218618 23	26.01744720936 12	33.91560693177 54	38.27186666870 3695	26.01420215082 1603	33.68232035114 1	40.61228429556 6894	26.59337038257 1802	33.92936158866 3795
BBL	4.73853611915 173	3.81123134633 94996	4.12646340474 302	5.65202074462 765	3.81607001125 42303	4.24054389765 467	5.59558178786 857	3.92510619822 865	4.12663622943 717
BCP	2.48807789844 526	2.23579912678 632	2.17958627540 165	2.68120836762 441	2.22687237007 788	2.17049471056 26102	2.58702703740 706	2.26071611808 785	2.18566406742 947
BDMS	12.94400523632 06	11.39679683524 46	9.29394334879 671	13.00765834293 6402	10.53609060632 35	9.2494179478501	12.35869697929 04	9.13730919940213	9.26036485771 794
BEC	29.744973700378	22.91897220157 32	24.50112646366 1198	29.716115927081	20.95039983267 38	24.65226585567 03	35.26952124377 24	22.00182282147 94	24.56200558789 9698
BH	3.55647595645606 97	3.70566955191 03102	2.23208061981 931	4.23665249219 293	3.52579701060 924	2.28718781237 142	4.79393868477 97705	4.28848643654 993	2.23549731579 415
BLA	14.91026492234 96	10.53829321877 8099	10.15721897923 79	14.63025905158 701	10.45232754326 33	10.456232869672	13.68440740147 899	10.79506853836 409	10.14531222362 94
BTS	1.15831458719 026	1.29924846621 037	1.20980590079 854	1.19509345716 872	1.34352288236 426	1.10421825097 51501	1.40619318782 909	1.29824331344 01399	1.21000839099 99
CBG	28.55723946630 397	28.32061786671 46	28.35341176605 398	28.85936352613 21	28.07234542072 903	28.85238334422 297	28.51951024435 59	27.91252023355 529	28.17609241143 81
CENTEL	8.12705630704 452	5.42551324890 481	5.18071267098 466	8.89986834009 628	5.9336762251134	5.43860449805 438	7.83357065774 56605	5.49961049684 318	5.14494214303 903
CK	4.37157285907 791	3.72796290508 607	2.99494181487 92697	4.60328008814 865	4.26571300534 0295	3.12846957550 779	4.43323635340 2907	2.99018289305 061	2.99118289305 061
CPALL	3.31642994452 375	2.57032697809 611	3.12608974312 056	3.27669186229 988	2.72444026177 513	3.21636985403 144	3.31206036769 87095	2.51988676263 423	3.12150083725 478
CPF	6.78117191012 191	5.57970002949 761	5.74114636997 483	7.96477260178 481	6.08081979180 935	6.27099177536 2599	7.86214675268 501	6.05745810535 417	5.75057519944 129
CPN	10.91856148683 09	8.49244077896 041	9.84655640727 09	10.77201362736 5599	9.33058308341 252	10.11595618218 57	10.58840704424 99	8.77845154838 578	9.79691441316 8361
DELTA	2.31293725301 507	2.22979156001 98297	2.17091853216 919	2.49862767121 938	2.15285042131 983	2.28928265924 629	2.28995718669 83397	2.18658618374 706	2.17184835245 199
DTAC	132.4059171083 14	89.713625981491	100.09986535848	124.31392084198 1	86.038591336766 1	98.677181935866 6	118.128270994 367	93.724109797626 71	100.25206556 35
EGCO	6.48639209125 946	4.68582167142 988	4.72752674276 237	6.80031115117 28305	4.48284030272 118	4.7326325120425	6.73131038980 9241	4.86066341325 797	4.74635701464 331
GLOW	2.01899231912 655	1.78761036613 957	1.92735646430 626	2.19950163531 862	1.88450912060 335	1.93574051362 07002	2.05916586437 566	1.81276367064 21701	1.92659547226 3101
GPSC	8.58752959511 084	5.89753417495 883	4.79598247928 526	6.45878807464 54305	6.60312125138 54705	5.33063096439 061	6.94680483783 467	8.39499552313 451	5.14683382159 679
HMPRO	6.97754086713 9929	4.86429591794 587	5.64050650111 3371	6.36889303252 227	5.10412537485 331	5.25274639011 272	6.74808817220 391	4.70638851447 989	5.62667537962 826
INTUCH	29.346248027177 204	24.22985412596 48	32.04581033687 01	29.480792786147 298	23.82228292938 666	32.62734373826	29.220609457660 903	24.38776120584 54	32.05850462608 55
IRPC	11.492642037828	7.9505774954 235	8.38697169741 699	12.297141329464 8	8.63690899871 2391	8.25015636818 762	11.139688495775 1	8.13599568022 585	8.42221825671 841
IVL	8.97676388254 605	6.95472903567 8139	7.39640268491 8305	8.95712101801 738	7.59078526290 803	7.43698719137 851	8.91935173169 935	6.75215154186 7921	7.37389789667 571
KBANK	5.87058715146 55295	3.92241726528 569	4.49248376809 53	5.86832505288 451	4.03028852879 894	4.56049337513 915	4.88514765133 874	4.27263734939 72405	4.50766323410 925
KCE	36.22497333122 48	26.46696945482 3904	34.59814198817 194	34.13493834583 98	27.22513107889 6396	32.76494432469 59	36.45719574418 39	26.28588423874 402	34.52664524374 04
KTB	2.84009476632 973	2.27105374800 245	1.86220394434 747	2.39234223906 1302	2.42139066867 973	1.8134203167 615	2.63342178272 269	2.23078310246 175	1.86223798676 399
LH	2.56649442416 93	2.08943082300 647	1.70336125897 614	2.40943306523 505	1.96309349412 991	1.90692726429 81998	2.53282979814 52903	2.16276977230 23403	1.71657887707 75001
MINT	15.045677678942 601	10.889994475488 999	11.444874403188 301	14.002278887 3674	11.084757182685 701	11.08599139572 41	14.210874680312 498	11.829581804 2277	11.41908119201 48
MTLS	3.689710107599 8395	3.47954333406 215	3.40382844353 161	4.00669547833 97595	3.55154195363 76602	3.71416722013 432	3.80484598592 309	3.27827553590 209	3.44121886668 18597
PS	1.75889298203 071	1.50558684958 508	1.72693507020 86201	1.8417567454 846	1.47744294129 593	1.92331075476 591	1.82866875830 1098	1.46568027935 6156	1.72749467991 0198
PTT	8.10068566545 819	5.75872829647 294	6.19289539749 441	8.31741893926 065	6.57915074925 2009	6.41631078824 89205	8.26280237088 029	6.21152782766 875	6.19296589357 337
PTTEP	42.090723347019 5	27.796081158982	33.902317608038 9	42.788407844521 096	20.40379622855 51	14.724763086674 301	34.278388958345 005	31.849885427109 197	14.138911276766
PTTGC	3.882743031501 8403	3.815314602819 7296	3.69544810197 676	4.92239759184 666	3.93371110853 636	3.61836030410 681	4.52721686022 948	3.59609349939 833	3.69172343875 033
ROBINS	8.70063788895 849	6.65283832581 7	6.14822857522 38	8.77092652813 908	5.92813803898 866	6.32918303988 679	8.56688702470 477	6.20082406079 362	6.14384379906 971
SAWAD	1.64834432093 236	1.48555935425 68502	1.54119480038 612	1.51718555133 092	1.49099077638 792	1.62336000786 889	1.63925811444 69902	1.59467181170 32002	1.54372842170 976
SCB	11.725044674600 1	9.25093779880 568	8.54379884098 4241	10.894451536 7999	8.20618295624 142	8.26100352548 451	10.9587018725 4	9.15268962676 968	8.52174013137 752
SCC	1.55528633296 999	1.40645429205 328	1.42417292833 717	1.71350930414 24	1.40606955341 371	1.48530988702 934	1.56531397310 945	1.43259918636 26099	1.42455380194 544
TASCO	29.184059079543 502	21.917601615265 26	13.665880344 3687	26.810145992 2547 04	21.462787012 9985	13.525295169 7294	24.917568644 11 99	23.422553012 4994	14.076833013 4246
TCAP	3.880428314319 25	2.969775337321 72	2.934504719 61761	3.912690359 15348	2.997661980 51595	3.127785417 49915	3.940025210 92955 95	3.054947564 79405	2.933112047 75726 03
TMB	2.054483054173 5704	1.934426113340 1899	1.71525446740 65	2.046067454 01098	1.967562631 9026098	1.814881984 3339201	1.946699426 8542401	1.849390716 03344	1.714099079 26059
TOP	20.398392749 104	16.778383750 7106	19.109435123 133	20.323710892 8494	16.035376432 6236	18.179144531 8365	19.931848277 6498	17.169164099 2026	19.030538672 864
TPIPL	3.288029961255 27	2.16321644891 55	2.31777796451 37098	3.11697703958 422	2.22432903810 535	2.19085730380 231	2.58692948804 218	2.16514370100 634	2.31981738256 371
TRUE	10.710194321609	7.48843121550 745	6.90304544273 939	8.91941602532 23	6.74161195056 26	7.15674204163 718	9.35997918744 326	7.29463819088 869	6.89928013471 655
TTW	1.29552461899 304	1.09132384973 58302	1.16936454461 509	1.37152974626 281	1.06542169696 63999	1.20215933424 366	1.30017197291 44399	1.09601495393 67798	1.17515346441 227
TU	2.61008056157 666	2.10711103996 447	1.95958037485 7399	2.37992477397 896	2.14535370820 81997	1.8630389675 179	2.78411229743 24	2.09559823637 1603	1.97766466251 388
WHA	2.16951360114 557	2.15619520835 523	1.83615953541 944	2.19742317823 667	2.10614898105 229	2.02594679763 725	1.99949393476 22798	2.16023875319 503	1.83491580485 521

Table 32 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.90409759479433	1.92370360732773	1.93115616403632 02	1.85592474330 08899	1.83400105805 281	1.90053779509 594	1.90670459419 83398	1.91817920307 463	1.92994486236 085
AOT	1.09271353841 818	1.07027091509 561	1.05349193481 564	1.06836664036 43601	1.05299931592 084	1.04468625895 161	1.09196404615 013	1.07051355329 853	1.05338960894 336
BA	0.943886426537 324	0.915850066184 223	0.91861293182 4304	0.95733449540 31859	0.91102619690 5378	0.95822037212 8327	0.91111149452 7052	0.91637007201 94949	0.91926372572 00029
BANPU	2.07481241821 936	2.02663183912 69698	2.01608235851 637	2.18787775097 572	2.09017568510 123	2.05876468842 941	2.08321733349 52997	2.03615422021 868	2.01534758772 84
BBL	1.22063805741 224	1.20949758767 209	1.19890131069 313	1.24099736341 905	1.22973999877 763	1.22966761612 97	1.22227231384 979	1.20940513671 836	1.19872746475 966
BCP	1.43117387586 861	1.39130274495 123	1.38945591075 76299	1.49936989762 972	1.42760061427 31799	1.3507445553 002	1.42090192647 23	1.39050418709 387	1.38939926780 821
BDMS	0.925483731427 956	0.916621718626 8809	0.88522171826 424	0.94433868798 7313	0.92651638043 1819	0.91571997417 4959	0.92790041670 05009	0.91498689500 3679	0.88592963013 334
BEC	1.95767983265 15599	1.88707338349 29702	1.90345990144 74	2.01501834787 111	1.88774911261 05501	1.90495532348 48402	1.96266540332 06299	1.88780594815 95302	1.90581950312 121
BH	1.43480879446 09301	1.41029332164 256	1.39631229562 979	1.47145321073 157	1.45513443397 656	1.46703205803 56	1.43129853646 56501	1.41265404700 95899	1.39615838542 05402
BLA	1.64980058976 658	1.62775368595 759	1.61796656571 47302	1.69521066609 647	1.69220918991 32699	1.69928492123 239	1.64780915503 36997	1.62682387023 7	1.61796372290 80199
BTS	1.01929594184 15901	0.99192595634 6062	0.98965799458 7338	0.98958000271 09399	0.96372224876 63111	0.96637243013 98811	1.01146797472 705	0.98894191694 6145	0.98936699241 69981
CBG	2.04956313497 814	2.11162296751 97498	2.06265398780 10197	2.17839945314 463	2.15693186175 495	2.20364196232 79397	2.08586920100 639	2.12007256848 412	2.06428666494 749
CENTEL	1.60090287664 441	1.55990104669 13	1.54811151910 246	1.58743947217 504	1.55236077037 33301	1.54798447004 298	1.60183423676 422	1.56761768967 522	1.54787518149 812
CK	1.39352034127 2	1.36559725275 06	1.38377337413 43	1.39126533604 982	1.37849504514 366	1.37821570504 318	1.39492308116 87099	1.36466221261 768	1.38335356672 385
CPALL	1.40029544540 288	1.37518598516 52302	1.35627571398 166	1.36909393101 113	1.35351220821 67301	1.34137397034 98398	1.40129896631 37199	1.38074155706 36499	1.35627359315 445
CPF	2.09593685290 39297	2.03901471628 143	2.04364970832 86502	2.01673653965 201	1.99919793602 552	2.01012913032 312	2.09579596883 532	2.03864628755 8	2.04355669003 32
CPN	1.28767591434 18	1.26141168945 212	1.24739346772 536	1.25322293313 604	1.20705601398 493	1.21724755992 007	1.28573033721 696	1.26413748653 153	1.24825468070 7
DELTA	1.82813198835 143	1.76538176369 47602	1.72720573090 977	1.81124784183 395	1.76298164091 423	1.74966073640 14398	1.83914613183 005	1.76218077519 07699	1.72723854849 7098
DTAC	3.25377254375 337	3.17956957036 826	3.16306895106 946	3.30333949943 518	3.23177430784 996	3.20799277382 086	3.27246058071 324	3.17817744110 23296	3.16246837015 109
EGCO	0.931459368797 3389	0.915088932034 929	0.91960791057 8794	0.90600047362 93099	0.90455031355 238	0.92324184008 7771	0.93900231705 146	0.91529162115 61279	0.91953338534 97689
GLOW	1.33121225178 811	1.30546878942 603	1.28782536717 981	1.32913549178 20601	1.30528091851 04601	1.32349016525 868	1.32810343384 14	1.30751640445 5099	1.28831836650 827
GPSC	1.51196808731 743	1.54930768081 123	1.60709414717 944	1.65457799267 47402	1.67399448947 58398	1.75493951409 31199	1.51052293885 317	1.56165520805 809	1.60698188520 937
HMPRO	1.47613898499 833	1.46569282048 69901	1.47095618614 241	1.47124710447 47	1.45967107506 883	1.47812414089 574	1.47476509963 375	1.46617968620 559	1.47091236092 1602
INTUCH	1.67993645683 22601	1.63823015389 37198	1.66214817436 331	1.69641355113 55098	1.69036509031 73	1.74960898102 48898	1.68892437280 389	1.64303889329 502	1.66203502707 224
IRPC	1.31897751083 814	1.31800242887 268	1.31883905305 574	1.30871800350 436	1.30755030563 45	1.33621085505 84701	1.31751472776 309	1.31640489546 723	1.31878855928 876
IVL	2.13459813590 22	2.10166653940 915	2.14974868921 196	2.15483998078 628	2.13962252305 194	2.18461775349 20597	2.13353615597 46397	2.10159587871 822	2.14996187217 28697
KBANK	1.60117315455 724	1.51709001227 192	1.46709030774 781	1.47858631372 67798	1.45554040965 769	1.44179608092 899	1.60580653639 563	1.51132265912 46284	1.46751590654 02799
KCE	1.46122500632 743	1.44748006296 28999	1.46460947567 549	1.47056902044 46902	1.46088061170 12298	1.47709973538 526	1.46118707803 888	1.44628701179 921	1.46489821624 484
KTB	1.30744975105 734	1.24138144882 839	1.18371912049 22901	1.25916481902 671	1.21512771612 10098	1.30925615461 998	1.24195843161 533	1.18356134691 268	1.24662092556 249
LH	1.30556096369 29901	1.27922384761 859	1.24739973983 289	1.26253965841 833	1.22366388852 182	1.24834413060 578	1.29933303711 06802	1.27866674810 167	1.24662092556 249
MINT	1.62405270477 21598	1.61673410723 11598	1.63578170904 057	1.64421398112 959	1.63052655846 70802	1.63720114167 06598	1.62509755729 617	1.62110469147 096	1.63563961730 59298
MTLS	1.04791418683 309	1.02456077815 963	1.04513406533 33	0.98554483603 65501	0.97942973464	1.06100974820 275	1.03433038179 04599	1.02474746213 567	1.04566717829 037
PS	1.37463209253 635	1.32240064651 06601	1.24703202997 701	1.33772133168 196	1.28367010809 042	1.28128338743 21998	1.37898262225 328	1.33662722891 483	1.24752041275 478
PTT	1.81750034333 651	1.78507827199 655	1.85764599993 114	1.79071570073 74601	1.79796440966 895	1.81200284388 502	1.82098142688 651	1.78295080062 12702	1.85735274974 001
PTTEP	2.60474771176 357	2.58565046674 688	2.49249815702 622	2.38389478120 967	2.33648347941 161	2.33080103318 86697	2.60062476183 701	2.56426363384 54	2.49430670890 22
PTTGC	1.83812095864 92702	1.81774862105 053	1.78368726339 697	1.87187477103 033	1.84887653328 51198	1.82515862796 32799	1.83151625563 918	1.81819580318 86098	1.78379543290 803
ROBINS	1.60377410839 49502	1.53556777314 4666	1.47995713236 1601	1.53853167763 726	1.52388611631 60902	1.51426344701 747	1.60078652122 66301	1.53545636569 669	1.47970642496 2598
SAWAD	1.50565544933 364	1.44392843759 174	1.42572106954 775	1.42383072810 998	1.43682368807 499	1.44701012597 151	1.48997472501 522	1.45345743925 198	1.42658442579 669
SCB	1.50910720973 511	1.46211975392 972	1.46986850985 72201	1.49697756044 172	1.46868754568 476	1.48488405737 893	1.50967617404 488	1.46165326413 328	1.47041095425 21901
SCC	1.13159050918 643	1.10458308741 548	1.08933755057 643	1.13966063985 847	1.10875584875 565	1.10517051310 945	1.13344237561 098	1.10538611654 996	1.08942534093 771
TASCO	2.40170902808 552	2.39560713806 928	2.36121151531 699	2.47132629931 931	2.47378357237 415	2.41245226792 849	2.41355290293 40597	2.39327855902 76997	2.35834993153 613
TCAP	1.16106422192 31602	1.14284442382 243	1.12392841576 444	1.15684574648 223	1.13396274231 3299	1.12672109002 097	1.15471048929 792	1.14308304363 733	1.12385044868 704
TMB	1.44079821799 706	1.43825011498 17701	1.41031342317 763	1.43070373309 509	1.43035044146 565	1.47763641441 881	1.44116968427 661	1.43912570621 372	1.41068496899 271
TOP	1.52630010662 168	1.52524916762 739	1.51652806232 786	1.52131056001	1.51536590232 484	1.51126873906 189	1.53397657297 785	1.52572615114 861	1.51638970524 585
TPIPL	1.84805437264 14702	1.77147286105 00399	1.74937226847 41302	1.88882164707 25399	1.77507601429 80202	1.73046790551 32302	1.84337164405 232	1.7774157975 945	1.74647312807 742
TRUE	2.13890445396 355	2.13591862287 57503	2.16726104024 52	2.25087429258 559	2.16530832743 024	2.24047253389 849	2.14380641959 122	2.13895148463 265	2.16537242834 282
TTW	0.863769299858 125	0.84295934308 175	0.83546636582 5774	0.85523292172 9447	0.82419142670 31201	0.84019940938 096	0.85688611408 3045	0.83969685719 9828	0.83574794275 61
TU	1.36340815346 661	1.35240770054 91302	1.36313235755 85002	1.36891101811 384	1.35686498583 498	1.36671771209 604	1.37451211057 665	1.34637905379 281	1.36198577126 408
WHA	1.58184952883 25098	1.56128901644 928	1.53755905398 23099	1.69735290430 44702	1.65368209171 773	1.62935823480 97297	1.57103719513 313	1.56128308920 175	1.53754996908 299

Table 33 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.81462856962193	1.82816155903451	1.80835619018429	1.79778459217338	1.81274548637978	1.83623783133798	1.81394710392101	1.82389520580112	1.80738864860677
AOT	1.10629491784613 01	1.09163755882515 01	1.09125793536997	1.11562940662765	1.08160927800346 98	1.06670879273055	1.10833389471441 01	1.09881202351068	1.09103812196186
BA	0.92271344871008	0.91112315624937 391	0.90773028825009 92	0.9858050784007959	0.92999673239057	0.91210160406518	0.91585201323591	0.91869519287299 71	0.90588451984892 9
BANPU	2.06148644476321	2.02441238257689	2.00055170349330 02	2.15560692521164 96	2.08395842726277	2.04648915340076	2.06403952862221	2.02299283029072	1.99922493557214
BBL	1.21347035788291	1.20779917016655	1.20094437491056	1.23687254869183	1.21589533194213	1.21665819838127	1.21378667038219	1.20798792496925	1.20052781228022
BCP	1.42778935881304	1.38812333277449	1.38945591075762	1.48717003939372	1.42757706905354	1.3507445553002	1.43198514793425	1.39163530149396	1.38939926780821
BDMS	0.92163469104215 6	0.92005386251158 7	0.88554481691662 4	0.93970427010097 7	0.92282070812726 59	0.91571997417496 2	0.92127492124671 11	0.91669082307029 7	0.88595122528094 11
BEC	1.82460724353941	1.76477732349955	1.78054965548726	1.89667534181506	1.80540378350432	1.79244374373266	1.81647359078508	1.76846963717446	1.78232292352209 99
BH	1.44813656789253	1.41731171037061	1.38840755715225	1.49604267069176	1.50925914345552	1.48670842846433	1.44714390483193 01	1.41591453626333	1.38896094526428 02
BLA	1.64556095170944	1.62801633819716	1.61800098226495	1.69537755182560 98	1.69036212797853	1.69928492123239	1.64631363398829	1.62838280827855	1.61804130360461 02
BTS	1.02491801259379 98	0.98098997195616 69	0.98002129558018 29	0.95834977883228 39	0.94199420050576	0.94325247148621 6	1.0247716232824	0.99544953952974	0.98011634658754 41
CBG	2.06718249765317	2.11159417989955	2.06265398780103	2.18012486659695	2.16117899283 68	2.20364196232793 97	2.09094628142637	2.11571898479937	2.06428666494975 002
CENTEL	1.59563543544661	1.5675458981498	1.54953771667338	1.60202753685097	1.55939958377967	1.54009166215783	1.61803149608873	1.57327398098702 02	1.54957759833142 02
CK	1.44571188734177	1.39880376402288 002	1.42055428576233 01	1.51222305484985	1.41739012565521	1.40685309719323	1.4424933933905	1.40110913345276	1.41945127587944
CPALL	1.40138078440098	1.38048018056104	1.35627571398166	1.36502141214557	1.35230874133689	1.34137397031943 98	1.40194400731792	1.38047918376529 99	1.35627359315445
CPF	2.05619080722012	2.01665829320019	2.03308162087506 97	1.99860594933004	1.99311363394649 99	2.00674277389189	2.05753946891453	2.01665566913517	2.03373767234256
CPN	1.28704051671508	1.26132807919184	1.24793946772537 01	1.25279451932577	1.21677810687959	1.21724755992007	1.28841623945147	1.25897926017983	1.2482546807 07
DELTA	1.81217712876931	1.75659952724669	1.72720573090977	1.81277850204852	1.75408761557453	1.74966073640143 98	1.81051372143604	1.75741806029354	1.72723854849170 98
DTAC	3.22320498807425	3.16355081054365 96	3.14703028165941 05	3.21330838942096 97	3.17112973936847	3.19897912770863 04	3.20628732772426 03	3.16053686229281 98	3.14686442341704 97
EGCO	0.93349471609912 3	0.91472936797641	0.91992603802293	0.905810131525427	0.90473494107469	0.92322663715016 8	0.93606757624749 29	0.91470407728311 041	0.91984690073535 721
GLOW	1.32128054575012	1.3155672430816	1.28769841125969	1.32914464759925	1.30063317391569	1.32399278441365	1.32814058994144	1.32178507611143801	1.28794606152824
GPSC	1.50407608379371	1.5587557537231	1.60709414717943	1.63435153378352	1.67621748287511	1.75493951409313	1.55610712469929	1.56175598600102	1.60698188520936
HMPRO	1.47547925622988	1.46237498436579	1.46705084880686	1.46424913780333	1.45943933779618	1.4773366988926	1.46890019247063	1.46220308947897	1.46701518305923 02
INTUCH	1.6775692799486	1.64146441250529 97	1.66214817436331	1.69913122085093 02	1.69321440058014 02	1.74960898102488 98	1.68912708083711	1.63743140639966	1.66203502707224
IRPC	1.3653319688364	1.33898322523073 99	1.34051692353022	1.32600673921919 99	1.32148307341772	1.36152643628560 99	1.36664032108454 01	1.33960563534970 99	1.34037603778577 99
IVL	2.18559990955441	2.12783067685791	2.15198004194842	2.20185828167677	2.16067222349677	2.22432428827688	2.18069635476538	2.12721505515966	2.15126711114794
KBANK	1.54046730631606	1.49269935176835	1.45461306096048	1.44482541978705	1.43410558128729 58	1.41281519090922	1.54943699131333 01	1.48565523746487	1.45396472467276
KCE	1.46304086920082	1.4559381851249	1.46104412125131	1.47231408118313 99	1.46387021792534	1.48428204211349	1.46275583969877	1.45284187062975	1.46129784537727
KTB	1.30988412017871	1.2484246099458	1.18368410741325	1.2929574397736 2	1.25318205874 02	1.2152177161 2	1.31370208966055 98	1.24955618971621	1.18361518285118
LH	1.27207200797828	1.23797209343132	1.24129573716828	1.27437971426702	1.24495373290371	1.25048991242369 01	1.26737993893578	1.24024222048159	1.24127639101265
MINT	1.62425823225683	1.62265283016304 98	1.63225498942508	1.64771921034027	1.63451773601995	1.65168298335513	1.62282139146541 02	1.62332376828145	1.63175184423766
MTLS	1.04593616882002	1.02450747567792	1.04523012556753	0.98987526295038	0.97916827559893 32	1.06101552326862	0.98861869243495 9	1.02476001219786	1.04586779448961
PS	1.43457369553193 01	1.37268764647179	1.29261607847511	1.26959746004564 01	1.22323998501784	1.19948585890749 02	1.43434652978877 99	1.36993358640106	1.29197015000509
PTT	1.80820166949364	1.78445652169481 98	1.85811091379247	1.79010337278178	1.77935679319732	1.80983466252684	1.81916609527055	1.78415048725758 02	1.85785001879350 99
PTTEP	2.38535669003008	2.35425784241586	2.3545881794154	2.30413821047528	2.26858812103304	2.28194878154114	2.38676265863356	2.36105533615591	2.35152347134192 97
PTTGC	1.8449188042039	1.81857993974567 99	1.78368726339697	1.88206296748655 99	1.84698673170305	1.82515862796327 99	1.84842067287403	1.81846873078937 99	1.78379543290803
ROBINS	1.58645290063491 98	1.53972742668022 99	1.47995171323616 01	1.54324769095686	1.52286708434418	1.51426344701747	1.58657081677944	1.53724233398125	1.47970642649625 98
SAWAD	1.50653629311463	1.44058179965224 99	1.42570955373113	1.42989607581497 01	1.43031760408518	1.44699469721568	1.50871365528654 01	1.44961587067344 99	1.42661545194857 99
SCB	1.50767908219813 98	1.46159975695676	1.46986805988572 01	1.49653702939484 01	1.46809654110078 5	1.48488405737893	1.50904717462933 01	1.46178182207325	1.47041095425219 01
SCC	1.14183927632374	1.1048401829009	1.08388070121065	1.14710307585964802	1.10696827858805 01	1.10562797285483	1.14916434439759	1.10541082777560 98	1.08432783572591
TASCO	2.50316349077249	2.43696615104055	2.34814438706709	2.75604161306665	2.69476669824044 03	2.70620609878536	2.50421574622235 98	2.44378847494327	2.34708545185947
TCAP	1.15440685241599	1.14179377771556	1.12392841576444	1.15500181754537	1.13366555657497	1.12672109002097	1.16314886111008	1.14385412870337	1.12385044868704
TMB	1.44277644502699	1.43808351555241 99	1.41061849933579	1.43848638623151	1.43093997470562	1.47027610967221 01	1.44197468620869 02	1.44064637144638	1.41078971032866
TOP	1.52161911895831	1.52539749683372	1.51652806232786	1.52114423998087 99	1.51532437524253	1.51126873906189	1.52232078643768	1.52558408217961	1.51638970524585
TPIPL	1.8768612878924 9	1.84538877065531	1.81644037469766	2.07823417528338	1.93161353187831 99	1.75358442605508 01	1.90557300017457	1.81245866843389	1.81479742543533 01
TRUE	2.09327537613436 96	2.09350488311439 7	2.14525921890348	2.19532587779463 03	2.15461180334012 03	2.12875894735124	2.0968791836562 1	2.0868909675906	2.14154122361088
TTW	0.87208042194530 9	0.85406997125799 3	0.84059971843526 9	0.83714040251080 71	0.81868598643222 19	0.81937714157788 81	0.87085691526394	0.85343219638210 4	0.84063776260177
TU	1.36454204908185	1.35033790144512	1.36314178253336	1.36481612778864	1.36155824066233	1.36671771209604	1.36696416327657 99	1.34868871419223	1.36204601275346
WHA	1.57077562902020 1	1.56194814469323	1.53533071328618	1.60266070582968 99	1.65964446088236 02	1.6345915836 64	1.58285563393354	1.56270667323879	1.53538364447679

PredictPrice12 : ATR (Average True Range)

Table 34 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	31.3542327052772	30.8302185554668	28.2055742712174 97	31.4726310622187	30.9077423560999 98	29.6946183974 88	31.6188147301648	31.2417205487175	28.4597773998837 02
AOT	29.1662891459333 98	30.4716324434777 98	16.8002428999609	30.0561515956612 02	29.7576392308658 02	17.2375130197527	29.8857107979099 397	29.1375599923637	16.6961467236115
BA	1.56460681195244	1.65491817684000 92	1.06633085894163	1.79831458717404	1.44870445523601	1.44161620232675	2.35384657199942	1.56583681603402	1.03726577443045 99
BANPU	40.4237871290036	40.7075426724314	15.2992792737269 01	43.9573327464032 96	35.1393668559356	17.8180167312674 98	39.4914131816739	39.5978452453911	15.2931859260272 01
BBL	6.96167494510705 1	7.93754961429460 1	3.63995149871671	8.42367146872597	7.13308513519534	3.96171401372067 94	8.11394253008738	7.99042829927796	3.63913452905437
BCP	1.83694433853737	1.75345844224718	1.38836579845208 98	2.05617890318742	1.7814729796763	1.43602781255289	1.92856033121543	1.72522959317372 98	1.38999670714609 01
BDMS	15.666671367638	18.6451638178714	6.52636070749322	16.9148055022941	16.5262259596334	6.38044072069013	16.0388162177 57	16.2882896207679	6.3554862941272
BEC	44.1921614904919	45.4936864620912 96	16.6033390197257	39.9079539488197	39.6620948966 44	16.8673909641797 98	40.7559648353598	41.6184022105622	16.6380841075 35
BH	3.12931354894966	5.5880499518670 1	1.55324544998131	3.78238248061558 97	3.6757118293056	1.56013191262406	3.92583511152275	5.6200541964488 1	1.54532071789098
BLA	15.9692010603678	16.6960907908192	7.63493277024155	16.9383912372653	16.5664292575688	8.70301133956143	17.1487772266452	16.0526396566823	7.61098150517549 05
BTS	1.52698676426601	1.7303798631339	0.99300754685736 1	1.87495748225706	1.81713500705442	0.97835830756270 5	1.50421059617918	1.64307553583727 502	0.99246978379177 8
CBG	28.3984419467731	28.5610657068952	27.0218171678365	28.3518214921264	28.335125551724	27.8618735542259	28.4365717890 35	28.3174319388173	26.9255695558408 02
CENTEL	9.6803153202339 6	9.6701812184274 2	6.6790347630603 5	10.2112356141081	9.7010067914424 31	6.6241590213614 1	9.5774534714691 6	9.9470526177012 89	6.0146287099781 8
CK	5.4255394561544 4	5.8404831917825 7	2.3681061746944 9	5.2475111335248 5	6.7004466434922 5	2.4689516689727 2	5.0580714083479 1	6.1281591345116 3	2.3424694226588 603
CPALL	2.7125969733468698	2.7105166465936 1	1.9122350164549 8	2.9793673833313	2.6741228832085 5	2.1271838426678 102	2.8672403950774 1	2.6929173772907 03	1.9117625691063098
CPF	3.2407169760081995	3.3531732268638 37	2.8946323963251 9	3.5925945329110 8	4.5832708004022 2	2.8693273211956 4	3.2787370275842402	4.5311146355268 6	2.8927578962457803
CPN	10.916941423228	11.124123851980 9	9.5803466821998	11.1596303724282	11.0601223692599	10.2878818254061 01	10.8104055106237	10.7196287740922	9.4600949006643
DELTA	2.6919603745067 3	3.6557450679329 97	1.8414774275744	2.6020279708745 7	3.4824853208711 1	1.8379753907347	2.5587367220042 3	3.0389890950681	1.8435303344261502
DTAC	159.867321852153	152.888762176359	71.8701525189997	159.417196209917	153.852410649684	77.2823141743203	157.155541292118	158.556875633518	71.7721477006085
EGCO	8.2692061900713 81	8.3991546466839 5	3.7326186999354602	8.3168612830515	8.6803101260957 99	3.9216506098963695	8.3479661362617 7	8.7106673088745 7	3.7266361264214 8
GLOW	1.2837489790134 6	1.4770329264 81	1.3647330429670 8	1.3626460074739 1	1.5081211361233	1.4272874923641	1.3030891980350698	1.5284335703614	1.3653134628421 6
GPSC	5.1536733788607	3.7866228799469 2	2.2326443154704503	4.7259734532843 3	7.0481338979329	2.5651703047854	5.4371845047665 11	4.7802462503656 1	2.2418932148897697
HMPRO	6.1005008329720 8	5.4938886040311 6	3.5895285594292803	6.3723847977219 1	6.0729618823952 1	3.8017446896150 9	5.7585150749874 9	5.5479570299545795	3.6017652305708 5
INTUCH	27.212127641511202	26.151969881869203	23.781473232736	27.7299217395275	26.4546676763069 97	24.9684990809614	27.8729938255921	28.2207159766956 97	24.0691727043094 98
IRPC	11.969354233923	12.983666434779 5	5.5168709033270 9	11.2528554390443	13.168758520485	5.3554693643193 7	12.8242255265316 98	12.6548022743562 5	5.5611231179906895
IVL	10.2548058352186	10.3857617541142	8.4041786054137 1	10.4842874366101	10.5197995019151	8.9251304746340 9	10.2851117293429	10.1766658634191	8.2906627573033 5
KBANK	4.3250833851026 3	6.1307219661063 7	2.4601425311764 6	5.1511107177199795	5.4388593634276 2	2.8051366939761 2	5.0631347867329	5.7501677151738 7	2.4585160757611
KCE	36.951739637906 5	36.1308257464593	19.7474036418929 98	36.9089442368683	37.7790079676346	19.8226440347314	37.2032693774950 05	37.4949164254916 95	19.7625957154743 97
KTB	3.207760662539	4.2798234506478 9	1.8139017500458 3	3.0200566719230 1	4.1940312116493 6	2.1052877239779 8	3.9214277599744696	5.1550375983233 1	1.8054198734977 1
LH	1.42895487539932	1.53187877257912	1.2929375022434302	1.3174983841055	1.5212157933368	1.2886680300773 6	1.4059875107960 7	1.5285363224171 9	1.2918998748962 1
MINT	23.775053561035	22.9623812935228	10.6923332415564	22.7198277102601 98	22.4501767962274	11.3137448656203	22.7249344717 35	22.9595959388307	10.607156829696
MTLS	3.5387490806969 1	3.4981275007703 8	2.2918090859121 5	4.2405179002367 51	3.9743393287480 8	2.2360794437794 5	3.8751722096745005	3.5985483071356 6	2.2968614440169297
PS	1.8161133558362 1	1.7287254804420602	1.3525722696819 6	1.4508311780965601	1.7511926738497 7	1.3395723027092	1.5224607435231 3	1.7307773006514	1.3562806912074
PTT	6.6384136130182 5	7.3864227817656 2	3.8880005173739	6.7745944119802 3	6.8987666896097 39	4.4591970842446695	6.9160984965673	7.0378970910095	3.8863139895385097
PTTEP	35.3252092728803	33.3394496878581	9.0427154235827 1	29.4080127355754	37.9912638396082	9.1343175649905 3	41.4022192554 05	36.9398255651010 96	9.0554911587986 99
PTTGC	3.8808353753306997	3.8973139149642804	2.8030052876054 2	4.3341607559856	3.7874293723123 7	2.8577640765558	3.7758348393812 1	4.0005151520146995	2.8075728607050596
ROBINS	9.1519521206017	8.8610766296815	5.0121750997280206	9.3011000289968 6	8.2597614894668 5	5.4983866878199	9.4201631020858 5	9.3904923236842 56	5.0056204962293 4
SAWAD	3.2941529437878 4	3.5532417318498 8	1.5346553075073301	2.9291448206985398	4.2581418668627 8	1.5975983650818 5	3.3208562279948497	3.1244341528745	1.5341886536107 3
SCB	10.7500232576234	10.0914300569017	6.2421941428994 1	11.8094206050687	11.4991988818881	6.1905513526129	10.1557902048345	11.3089620505232	6.2660258771441971
SCC	2.3187984961742 4	2.2186195357745 5	1.1609207479286 4	2.3380431360949 1	2.2938562326248	1.1968204019101 8	2.3136268074041 7	2.3751476577487702	1.1605863649821 1
TASCO	21.6399231839536	43.0830585552894	3.3509135835119997	23.1690326711189	42.3699223539423	2.8416450125933 19	22.1354362433635 02	49.4273180003935	3.7213018696490 68
TCAP	4.6637942137903 39	4.6706143501611 8	3.0491808172647	4.5995638794115 1	4.3191033666540499	3.4997501011294405	4.5184772716904 8	4.4337609884416 9	3.0582613037631 5
TMB	2.0444546856445 5	2.2679899232786 6	1.5924779775616 1	2.2301965478462	2.0789223503211702	1.6575145589873 2	1.8741343495676 16	1.9803886252874099	1.5944851565595 2
TOP	18.658975137183102	18.4810827072099	12.0312168521386	18.9733641237348	18.6879470358414	12.8053746898103	18.6618362284592	17.6570006988231	11.9222228731358
TPIPL	3.7666842393473 7	5.1595278852215 6	1.7814339519703501	3.1249135411065	5.7362479582356 2	1.6569112780264 8	3.1230137635403 5	3.0167651434841 9	1.8286633345972299
TRUE	4.7631731411282 5	7.3402579675935 39	2.5173960640463	5.3374123276956 5	5.4330980304441 5	2.6342172512992	4.2083253857695 6	6.9665937251970895	2.1529554816062
TTW	0.9907761295533749	0.8995987499263 66	0.9195574582061611	0.9938028224185 51	0.9807256985813 6	0.9437072528066 26	1.0160632346096 2	0.9171915669665951	0.9206802296802 93
TU	1.7077500503235 2	1.9436645297804 7	1.3560355926274 1	1.7690559154660002	2.1451121424251 3	1.3696501453110002	1.6567589214636 8	1.8138178302090 12	1.3542163261630 6
WHA	2.0447638041803997	1.9052680158 51	1.6321975741917 9	2.0706686984753 1	1.8273097657169 1	1.6940279938835598	2.0381851447270 8	1.8994161887210 9	1.6329017929045402

Table 35 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.8210106655157799	1.7901207552645602	1.7255373684576698	1.811855974891 13	1.7906148540489102	1.722450090888 02	1.820737261655 11	1.7910530093452601	1.725868725186 05
AOT	1.064367010596 39	1.071395720302399	1.036519226751 98	1.061237438190 87	1.055764261331 8101	1.048158619808 4	1.066558508365 696	1.070614751537 82	1.036519686264 838
BA	1.352794692424 145	1.359374575546	1.136079276593302	1.107100407999 97	1.071127669598 15	1.161177310796 79	1.322244132658 68	1.277112711864 28	1.132369888843 314
BANPU	2.045841791823297	2.050500656697 34	2.007059193500 1403	2.207196818048 58	2.172230650631 95	1.989994982795 84	2.043744317345 77	2.065610012468 76	2.006796611922 306
BBL	1.281335463148 94	1.262968384701 08	1.22087379177 183	1.295474240981 87	1.293548861032 31	1.245448848194 8298	1.289093286373 698	1.260984215762 87	1.220823845265 5
BCP	1.376768992086 04	1.424517373446 96	1.340449836126 4099	1.353905756036	1.391723102148 78	1.307120217208	1.370479333851 8	1.428956723338 55	1.340537263865 51
BDMS	0.922451338900653	0.937881048547 78	0.876352440858933	0.956699776767612	0.957627322434509	0.913654226382 4969	0.935869106607 7949	0.931127090659 741	0.876616071037 244
BEC	1.814881745184	1.762106677903 7	1.670272394976 68	1.729273654027 55	2.011693849376 54	1.748216055434 7599	1.831221371082 95	1.780670487226	1.670624982394 63
BH	1.448613704089 14	1.460094039606 2	1.408999302898 45	1.474563001516 55	1.513011725255 71	1.474401721367 08	1.435142358398 56	1.500783771749 91	1.408860684278 1
BLA	1.676390202557 19	1.679238535725 349	1.639533217951 5503	1.765452927761 26	1.780415880028 18	1.763322665567 7399	1.649426231712 69	1.686732865836 506	1.639387790784 8302
BTS	0.977056689074 431	1.000277143365 51	0.924155171236 919	0.968068874748 457	0.981436832801 739	0.967610196596 386	0.956692542378 119	1.029052711293 67	0.924161379804 729
CBG	2.008356003284 3	2.017356078111 93	2.019917831856 92	2.056420209271 68	2.067789098283 84	2.122855112626 74	2.005314926606 59	2.003979585222 55	2.022213436061 86
CENTEL	1.544731725830 69	1.563849168532 898	1.520503082073 75	1.530108213781 93	1.505533155321 16	1.546339588304 39	1.546527813099 96	1.560886941778 86	1.520110269806 77
CK	1.334276209742 5198	1.372508343694 3402	1.330792096251 6	1.383097656326 79	1.392868490875 668	1.333612139792 22	1.334000058095 73	1.378483741266 67	1.330707720872 3201
CPALL	1.399235412178 966	1.376802560863 3798	1.320106618500 03	1.321450293232 57	1.317160477313 71	1.313590189117 59	1.390682584040 44	1.379141214190 715	1.320012010637 847
CPF	2.055746129598 49	2.100108026984 14	2.032946588837 2198	2.028257934164 51	2.027230259684 68	2.031151070837 503	2.041770594963 28	2.086072739029 24	2.029545826180 65
CPN	1.254540430078 48	1.233555576912 96	1.216866404686 58	1.279632961867 9598	1.282574037051 901	1.212114291195 28	1.251976658627 23	1.236479547069 31	1.216805419585 19
DELTA	1.862512191076 77	1.859856674093 1501	1.727085656575 35	1.860376793682 07	1.850146022072 4	1.707227629556 2099	1.856894091941 29	1.903207188923 1499	1.727910794834 03
DTAC	3.214034553646 296	3.208255403824 22	3.108923027737 39	3.368588331745 7197	3.331276464810 8	3.194179870324 595	3.202096573445 706	3.198187675453 1604	3.100710427833 19297
EGCO	0.930183449374 2361	1.011643222690 53	0.924688581662 694	0.921597225438 9551	0.947719979414 1409	0.913507555295 198	0.934697730649 0281	0.978879013904 424	0.924742395497 313
GLOW	1.335598171838 24	1.391075327724 31	1.261443867775 42	1.373106754968 101	1.389611485819	1.262832504563 9	1.328819928977 98	1.398559468694 78	1.261584169737 078
GPSC	1.788331487068 1	1.578372373002 498	1.490988069724 62	1.827983095150 8698	1.783061350742 66	1.787905005247 6202	1.881914190928 399	1.733523629414 55	1.493243119985 55
HMPRO	1.519856392866 37	1.505051034049 48	1.496452102929 05	1.509201922654 601	1.506880006071 26	1.474207185666 98	1.512724986531 34	1.503680056047 48	1.496580327981 27
INTUCH	1.660602533927 744	1.649241576268 23	1.621539028433 04	1.641426554973 899	1.636909791687 4902	1.612953654585 6	1.660278746193 2	1.650510970534 75	1.621785844353 29
IRPC	1.316300723004 1	1.329659755589 895	1.298723431380 69	1.360406959708 6998	1.378564843966 46	1.318890100070 52	1.312004248955 96	1.334856880739 6801	1.298910611528 96
IVL	2.148247159723 86	2.183505826670 26	2.117216811754 17	2.182728266921 87	2.183347057725 7303	2.094983203877 5598	2.184666442851 76	2.190246014298 4004	2.116642098100 3498
KBANK	1.493160551852 82	1.497800279948 46	1.393699917562 5802	1.417974050454 03	1.412116974672 07	1.394765146867 32	1.495568575538 78	1.510164992297 4298	1.393880303555 7101
KCE	1.510059495291 26	1.485743957909 84	1.457166459743 64	1.491762196834 01	1.488227802782 07	1.464025725584 88	1.510666257228 08	1.510316297214 601	1.457132343440 3398
KTB	1.223673709325 09	1.198559250636 06	1.159513895574 101	1.219489641357 7099	1.227863375459 7998	1.193242657198 24	1.207239561679 83	1.198771412946 73	1.159436327487 32
LH	1.276870933464 3902	1.343167972337 18	1.218979983740 32	1.226440645177 03	1.286102878968 929	1.199801865639 92	1.273386163957 95	1.337915735467 49	1.219090403274 85
MINT	1.636483540552 08	1.611319807196 38	1.588815708304 1299	1.674556336345 28	1.627573184730 12	1.589590641064 902	1.638187209621	1.611207063570 0402	1.588841563023 42
MTLS	0.937947100066 5209	0.919276691571 824	0.901821743315 822	1.007409122082 83	0.971671802596 3109	0.979204586939 845	0.947731427945 39	0.922187892658 539	0.902865046118 376
PS	1.240859184542 4501	1.392971983075 08	1.233854249442 24	1.264548896606 72	1.309939801927 18	1.254131241711 09	1.279566810682 45	1.391930108394 2899	1.239927872332 19
PTT	1.993197952120 87	2.018112974888 8202	1.879902461747 7198	1.917285169628 04	1.921842555047 57	1.820108745430 9	1.975377937762 56	2.016514917552 2297	1.880191366891 18
PTTEP	2.366229988706 44	2.355334394871 08	2.258296866148 8003	2.320544118245 21	2.257058357012 07	2.200151362567 9503	2.363299750401 28	2.359738439878 7202	2.258354981049 4397
PTTGC	1.841308894512 56	1.861617561357 6098	1.766102641500 1502	1.859477932078 3201	1.817570993161 9302	1.794174369538 64	1.802112288955 0802	1.854201118141 5199	1.766029176637 39
ROBINS	1.522660593486 499	1.558538075994 89	1.516018268696 21	1.546477161003 6498	1.553404132324 6401	1.463236202401 54	1.521781120624 81	1.557525166643 11	1.509574391022 8898
SAWAD	1.387022994523 24	1.437041719459 63	1.387127941166	1.429203095911 81	1.463179827229 77	1.432132909224 11	1.387000138425 4	1.410187111814 091	1.387242759131 06
SCB	1.458138728393 14	1.524705514619 31	1.442832473530 42	1.467719326056 8	1.474099506516 46	1.436532845527 83	1.445910886269 55	1.543790700441 34	1.443002551526 08
SCC	1.090752666570 7301	1.123020643592 22	1.072397560754 499	1.107237970082 92	1.107609354380 63	1.075393086557 83	1.102913985863 13	1.128431230237 601	1.072494315985 78
TASCO	2.432311833816 7498	2.372687730913 07	2.263006367191 45802	2.416167189459 46	2.420730402509 81	2.262779453001 56	2.434504780392 22	2.375239987891 25	2.262890232546 64
TCAP	1.190202048507 06	1.196383694782 75	1.139315873632 201	1.166642475267 29	1.166224748161 601	1.128890005551 66	1.187938219026 45	1.189204714369 21	1.139143364184 71
TMB	1.446655302308 51	1.446341197232 3301	1.397414937019 48	1.459132093403 15	1.487709968055 8301	1.483174216580 73	1.450295002998 79	1.445030907311 25	1.397527310477 08
TOP	1.548258369474 1001	1.570219275909 58	1.500909924147 39	1.557936929112 501	1.558807718669 702	1.510078321461 69	1.544252134570 999	1.562532571771 2701	1.501007716819 69
TPIPL	1.576004188909 31	1.694199889371 1299	1.584443218733 901	1.851754379554 98	1.832836130107 84	1.770596988706 88	1.590819001180 0798	1.680885719171 99	1.583220587590 91
TRUE	2.004430293910 52	2.094553853654 21	1.975236370129 1302	2.049936916335 29	2.130796767477 95	2.018033236978 8	2.007325784016 36	2.048314378555 53	1.975480433124 2202
TTW	0.831247272831 546	0.844608707846 691	0.820929763637 859	0.859834321431 751	0.841240793521 1339	0.834747446485 482	0.821952264898 21	0.838676618693 855	0.820538631035 8571
TU	1.360064139563 0801	1.394174907538 201	1.322099915724 201	1.370972239045 15	1.387267721318 2198	1.328893786811 492	1.353950018299 21	1.393952263113 35	1.322200191563
WHA	1.573337406571 87	1.566726848104 361	1.515882590732 6199	1.723791500289 6402	1.744144350068 08	1.594279672142 82	1.576523942485 75	1.566605827424 49	1.515619673369 63

Table 36 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.719889393920 95	1.730874584434 72	1.739205351527 1102	1.763311653178 3601	1.760715194710 4	1.736104007265 96	1.719503669224 87	1.730897704909 98	1.738603240481 2901
AOT	1.136779239409 9199	1.109281287987 97	1.027054111364	1.079718866420 6199	1.074196186016 6	1.034091494652 71	1.117584854314 27	1.109463669567 08	1.027256231341 62
BA	1.376411447936 246	1.342559281203 16	1.107917779686 55	1.304694150551 79	1.118002147302	1.032287806820 95	1.399178071506 58	1.311738408868 5	1.108030411806 03
BANPU	2.052039953834 7	2.052988465608 61	2.009522921932 18	2.183550310490 17	2.147405557336 58	2.002349503546 02	2.053170437815 61	2.064185957984 45	2.009297271157 4
BBL	1.258107270124 15	1.260721488305 65	1.217791900072 73	1.283719804883	1.273439022802 69	1.225222213429 078	1.258217551422 6202	1.261391343092 71	1.217755130700 52
BCP	1.368884906115 42	1.401223485475 1	1.340498361264 2	1.335822329001 55	1.366660486290 76	1.307120217208	1.367202356184 94	1.418941351140 73	1.340537263865 51
BDMS	0.933340885653 97	0.936214231234 074	0.876364734639 4491	0.937201872082 705	0.950971005928 7829	0.913654226382 499	0.926661422871 667	0.931510601087 2589	0.876616071037 245
BEC	1.715912197352 44	1.739403145288 04	1.677968070224	1.807824060853 5499	1.780249341970 24	1.688651813199 1601	1.710917055757 3001	1.735892701362 46	1.678331177374 3301
BH	1.504989549347 49	1.579913047756 29	1.425193926127 5401	1.567356582199 01	1.627223091056 68	1.45545017672 12	1.590701637088 7501	1.581534135454 21	1.420036749355 38
BLA	1.657225393082 91	1.686596508400 3498	1.639506563475 02	1.765512654577 85	1.779959925552 2299	1.763322665567 7399	1.670888628814 501	1.687857913316 54	1.639387790786 0102
BTS	0.964632123704 526	0.979117655877 254	0.910087277909 984	0.955037670234 9119	0.954664067354 375	0.976400192450 5661	0.963892433248 6411	0.977931269018 8601	0.910133098249 6559
CBG	2.006683382001 29	2.004102295062 69	2.019917831856 92	2.061303225007 59	2.071780027716 47	2.122855112626 74	2.001564122931 1997	2.011352930676 3698	2.022213436061 86
CENTEL	1.555621622492	1.566857126445 24	1.522929875935 07	1.533536299376 46	1.513174937241 16	1.547157483068 13	1.555604216820 66	1.558905431805 82	1.523240017701 67
CK	1.325305828655 34	1.393361545066 36	1.339215293878 415	1.372588517683 3	1.389211824629 51	1.347420279756 85	1.320773284432 1	1.363960156642 75	1.338308041198 35
CPALL	1.386842581846 19	1.380871999194 599	1.320106618500 03	1.339814356781 5801	1.331218418509 6699	1.313590189117 5998	1.384496555910 65	1.364373893292 53	1.320012010637 847
CPF	2.043791829367 82	2.049261424939 01	2.008883598868 2198	2.014674213434 3	2.053753574478 62	2.025692687071 09	2.042689655821 697	2.073185194896 08	2.007943061762 03
CPN	1.258365737980 52	1.234182894064 75	1.216866404686 58	1.279994059152	1.280706314897 01	1.212114291195 28	1.257450679249 82	1.235578376840 3	1.216805419585 1798
DELTA	1.857961019102 76	1.904024855228 5002	1.727085656575 35	1.836042681381 88	1.852257781165 2001	1.707227629556 2099	1.855719758576 18	1.911365385290 0099	1.727910794834 03
DTAC	3.154373291969 85	3.146292394904 5902	3.103169993338 068	3.330503392207 27	3.228168634688 22	3.308667323256 17	3.164704354526 3397	3.145440209581	3.099072226430 16
EGCO	0.944858193702 216	1.014025729543 59	0.925357251913 732	0.930805014074 7939	0.949370760292 01	0.913675244642 0159	0.931605583562 4179	0.978345440500 919	0.925406115486 3771
GLOW	1.363354144999 2899	1.399048424109 1001	1.261709864497 64	1.385153947025 58	1.382298064296 12	1.262802061370 96	1.388968254224 73	1.402724261262 7702	1.261842421172 58
GPSC	1.828230051367 19	1.567137079600 3399	1.490988069724 98	1.834504957823 9302	1.791671184383 18	1.787905005247 6202	1.521514882194 71	1.658663450858 94	1.493243199895 51
HMPRO	1.518511546335 34	1.509525150971 4	1.501602711616 06	1.515150609449 99	1.505790628759	1.477645032877	1.521666997938 1698	1.507916615252 78	1.501824767623 398
INTUCH	1.658253583972 6	1.650677105285 49	1.621539028433 04	1.641215429178 33	1.636890226578 01	1.612953654585 6	1.661318943539 7	1.645555788409 6002	1.621781497821 21
IRPC	1.351363352339 06	1.390482072927 92	1.308943082912 29	1.342506492595 52	1.395626817520 44	1.299738127047 08	1.350888356062 04	1.387128078711 102	1.309437594655 32
IVL	2.117696979972 91	2.117109831533 63	2.143418617331 76	2.149132540809 27	2.147750322936 35	2.157788188067 28	2.116390735407 95	2.112243877329 73	2.143361284062 1
KBANK	1.452381194644 25	1.476632747820 68	1.401791332732 06	1.411023307969 85	1.416181035602 0899	1.417642876826 299	1.445393994311 09	1.452905223031 31	1.401499606562 794
KCE	1.518520047702 63	1.513746318208 7398	1.459531586207 78	1.492495735857 05	1.490381516558 51	1.468828763725 92	1.518785979894 8	1.512178499073 29	1.459532735851 13
KTB	1.228496577652 69	1.199955573162	1.159550002231 1	1.215538812720 59	1.231437072819 27	1.193191736975 79	1.192944450896 902	1.200865405989 6902	1.159436327483 298
LH	1.274543112295 11	1.303075047045 802	1.224484119813 21	1.294420065942 99	1.308377242512 62	1.221056293422 7298	1.268142421650 5	1.342302195876 72	1.224533722415 75
MINT	1.621193248419 3	1.611380158382 11	1.603317908617 6	1.647281211732 73	1.636537259842 0298	1.622539873801 25	1.621441685567	1.614691467384 9	1.603006474472 24
MTLS	0.924521072020 619	0.933252551853 1081	0.901929508222 5289	1.010378388563 56	0.966515523721 06	0.979204586939 8519	0.930364912078 0889	0.921485453862 7051	0.902865046118 376
PS	1.346974398885 81	1.485430749706 75	1.281710021666 07	1.308435009365 83	1.418330092393 04	1.259383696566 78	1.405062033459 39	1.496729479669 1302	1.280086157126 21
PTT	1.991793932904 9301	2.018222547809 4997	1.880707701382 13	1.891507372088 63	1.914615612113 7299	1.820706552543 52	1.993024734734 11	2.023476038908 76	1.880979941366 03
PTTEP	2.219626001794 3	2.242400307176 21	2.213583854852 98	2.292706968208 9	2.253859760966 32	2.195484419203 95	2.208016439063 13	2.238617711655 003	2.213651737295 419
PTTGC	1.856689331606 26	1.845967420245 35	1.766102641500 1502	1.820166629041 9998	1.838300589167 37	1.794174369538 64	1.794192467754 32	1.888235780488 698	1.766215141042 68
ROBINS	1.522235773168 08	1.558388557593 799	1.516018268696 21	1.548072338566 04	1.552820140132 93	1.463236202401 54	1.522149673223 65	1.557178968705 3	1.509574391022 8898
SAWAD	1.386980981033 88	1.416184131749 85	1.387162485197 79	1.432787407411 3	1.448550642533 0001	1.432132909224 11	1.410253840934 03	1.412215899937 37	1.387100116222 45
SCB	1.457363055250 21	1.515829945884 59	1.442832473530 499	1.465801927402 87	1.466465061432 7999	1.436532845527 83	1.454823730897 8	1.531415727360 07	1.442991619796 81
SCC	1.094642446756 42	1.109129825810 19	1.072969991594 39	1.104934623079 26	1.104885871330 32	1.074710757928 0701	1.100582834112 099	1.107018227216 58	1.072827771755 75
TASCO	2.737861520418 81	2.669830102385 8497	2.802033253717 11	2.763786825319 21	2.780785053100 63	2.703551915835 82	2.778587020771 79397	2.697027353916 78	2.790826990517 58
TCAP	1.193668989407 82	1.196111829361 34	1.139351687363 2201	1.168185369186 9901	1.166554701249 63	1.128890005551 66	1.194457307703 97	1.195241293690 42	1.139143364184 71
TMB	1.454573904392 1802	1.447385784265 9499	1.394119393913 645	1.471862202692 13	1.491399924079 38	1.498895123008 98	1.432585825308 497	1.452542100096 303	1.394185688204 83
TOP	1.547188913333 705	1.557244778544 44	1.500909924147 39	1.561502890970 72	1.556291926546 62	1.510078321461 69	1.549390172412 64	1.561175242234	1.501007716819 69
TPIPL	1.652401517878 15	1.965726608722 72	1.595330054317 57	1.877137461658 8	1.931498543495 07	1.883608368966 92	1.699849061403 7898	1.887627625211 88	1.595759826470 53
TRUE	1.984433898949 29	2.020247029581 82	1.965667422573 1	2.076411209200 35	2.112180506672	2.016489828417 92	1.992139979441 34	2.011382624783 26	1.965566511548 07
TTW	0.858838373328 153	0.904629434216 226	0.824085039532 984	0.853281414338 201	0.888720352776 2051	0.791264979547 069	0.856177105655 4359	0.908282072976 22	0.824102399273 361
TU	1.350818624154 2999	1.395990919847 19	1.322095663979 6	1.387920490958 15	1.387920409058 15	1.328893786810 5999	1.352012023590 59	1.407046293248 2901	1.322200191563 701
WHA	1.562441007244 37	1.561308650526 37	1.510876856892 07	1.726654516873 78	1.738803094073 7198	1.603834411097 09	1.574753529617 52	1.579950724910 03	1.510918340876 24

PredictPrice13 : EMD from closing price

Table 37 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	30.01830880012638	20.814748930721798	47.7667125934 58	29.685417050970702	21.594318413965	40.1953923555318 95	29.76184912119 78	20.816033977773703	47.6309700275779
AOT	22.8127875946743	15.33582192272 37	37.840793355123594	22.2975947904663	15.065100936807799	34.46188869371 28	22.783339850 04	15.64699572512 72	37.82517213994 79
BA	1.602903054590666	1.4898070280132398	1.6033883074011799	1.630411571497427	1.44678558523827	1.710917967203 64	1.540623906715574	1.4948539755 1948	1.602314463123 21
BANPU	31.22532113222 63	16.2573405819913	80.068835977428	33.3051958569 76	16.752643651939998	66.7820715383552	31.722178252552503	16.16722854439 24	80.039819555413501
BBL	6.894047222711739	4.877704822788 75	14.9306661000168	7.21987766170684	5.00775124274679	11.500194045217 7	6.844157847154619	4.85384249195975	14.93359806055 18
BCP	2.190895281762 98	2.056163308585647	5.785492105329 15	2.428455306707 65	2.28322464041949	4.485026665871 04	2.152402524821 7002	2.06128531311006	5.812188705707 34
BDMS	13.020944517129 3	9.642936332667 9	19.99756949749 01	12.968524587499	9.175911155204 97	19.049629648965 2	13.35724777723 09	9.673541102723 97	20.030868351632
BEC	27.660184551822198	16.586093424842 9	65.411108998967 8	28.8304024175028	16.2259892165599	52.384358275814805	25.661764617341 497	16.751479944283 4	65.515108978114 6
BH	3.113599419113 41	2.975528900670 84	6.518907433491999	3.457206957924 17	2.589821716608 86	3.815168338327 1	3.304179387601 907	3.034499425731 8603	6.568731789736 29
BLA	11.469364096243801	9.697373404340 43	28.9170632995031	12.98169253512 14	10.364561528097 6	24.572129752228 93	12.759674115555 5	9.781181691886 25	29.058951518846897
BTS	2.467926987828 57	1.910628791031 5	10.340164816807 3	2.528354834024 66	1.952251950851 5	6.776376284285 25	2.636664158636 2	1.952559655959 13	10.331617932649 9
CBG	26.416281894281703	21.525004813784 3	27.696537532959297	26.960105033380 8	23.133663913845	27.461803666987 5	26.779346266027698	22.186340406661202	27.7037284920314
CENTEL	8.71278576932179	6.529108723872 32	16.302220129380 3	9.515333261428289	6.681545220251269 5	14.555016433171 3	8.856683625692149	6.479990026119 36	16.295543467311898
CK	5.948381179198 5	4.857364222046 38	12.208169050036 3	6.228190475655 55	4.68792149703425	11.144165799370501	6.07194425217 9	4.904952494598111	12.2300332925389
CPALL	3.389265801237 29	2.747935389584 03	8.422623352939091	3.365442684739 4	2.835510834341	6.709809428071 79	3.294876947286 82	2.738857824318 9	8.424565101703 28
CPF	10.561596551578301	7.455808676206 66	30.2931618382415	11.07792888269 09	7.596259864448994	26.54755183944 45	11.402718272931999	7.535485906346261	30.30723441613 67
CPN	10.148916878 126	7.328862729465831	15.166231090202501	10.3995447246 499	7.715909395168 96	13.422271358665	10.048079632 0959	7.430556207744369	15.137468217329
DELTA	2.809106111081021	2.377940801140 09	6.222656276436 9	2.834187321988 42	2.393895477781 22	5.035521818218 9	2.854448945960250 2	2.381643095962 79	6.221864347841 07
DTAC	125.32338793132101	73.23728333235 14	254.87479212382098	129.263523382226	70.3196837608732	209.669868347211	127.421873774341	72.359121313374407	254.863252915883 98
EGCO	10.11329492982 23	6.910894090545041	23.3576731945965	10.0528505549465	6.970276486554811	18.138090490215202	10.308056747515 8	6.985767973122781	23.509490470394702
GLOW	1.646719246884 5302	1.365971185989 78	3.766536689325 17	1.670788967226460 2	1.374115997526 19	2.774194876786 86	1.651486878488 83	1.359753911362426	3.767341585793686 96
GPSC	2.632537111836494	2.459299756709 19	1.932396191041349 9	2.585344533803459 7	3.663076526875 37	2.049169345904 13	3.568903553522140 3	2.402215601954344	1.926461721953430 1
HMPRO	8.036189242860 57	5.824156044072 5895	13.633450443846002	8.370493652980 19	5.774241304357 35	12.191245091820 4	7.969254276492 07	5.720698472916999	13.627856614248 9
INTUCH	28.3696740434466	23.385386478364	34.448155506777 7	28.51172598256 55	23.868157635429903	32.660653122215 9	28.676910672 9044	23.788432416645 2	34.567447472034 5
IRPC	10.403021132601 2	6.602178717595 20	26.521374854673 9	10.983339976932301	6.642641300242605	20.714728544799698	10.675093820419 8	6.871804296985 66	26.520728846401898
IVL	9.744305879087 77	7.626588100255 93	17.752543497352 8	10.008963253969	7.877604222498601	15.112845519283798	9.674221812746 9	7.711237497289 99	17.692897571831 9
KBANK	5.199329508022 64	4.103193179581 87	7.757443921790511	5.077083226741 24	4.096831744478 68	7.309912917702 1	4.654831089086 36	4.248345691512 42	7.747205916660 79
KCE	31.370706732895698	20.463169659528	53.810513182026	32.037787135965 2	19.691066175096 7	48.595248425122 4	30.720017793738 8	20.474043453662 3	53.767206714432405
KTB	10.963423520707 99	8.9488279548764 21	12.639233256761202	11.820710972207 2	8.785733039623 54	13.003918618970 4	11.317512779478 4	8.880300683112 74	12.680523960584 1
LH	2.605880081250 5	1.853491396144 37	4.808534939013811	2.575796365720 93	1.908170227139 68	4.302329098603 42	2.456199490668 43	1.878124850631 1201	4.807433381258 47
MINT	15.304364689044 7	10.684966959017	31.376898636063 3	15.300693647718 4	10.874147994458	26.564751938231502	15.739892619938301	10.92765999171	31.365824249807 9
MTLS	3.684735962356204	2.921366988739 07	4.544985168221 54	3.495771206106 9	3.416168280470 21	4.363499197378 49	3.714540138374 61	2.921624951092 09	4.538010180812 93
PS	2.414758757511698	2.690306259641 85	5.705934457071 02	2.446718246790 0197	2.589716358842170 3	5.746952923133 405	2.336936331759 06	2.683567312962 34	5.710806802519 11
PTT	9.484672633540491	6.612669930689 67	25.144201782209	9.860196925050 37	6.792592046395 2005	18.301502891820 4	9.634046956381 2	6.539350447443 59	25.148443523182 5
PTTEP	29.806077858244702	19.466475039225	51.134414477857206	27.456186952830702	16.5129898191567	43.202764212208 4	26.743770047796 7	19.436133139139 3	51.056418458646 9
PTTGC	5.281943484249 83	3.656266513948 38	17.484761298237 3	5.366729626616 64	3.705750353782560 2	11.913634135356 4	5.205295355780 67	3.701646479838102	17.566993726173 1
ROBINS	11.162136768855 8	8.456756722254779	18.643648167306	10.952400314435 7	8.536863541950 08	15.573456505528 1	10.954809253805 1	8.464015279371 82	18.702324550158 8
SAWAD	3.638739671751 95	2.814787035737 91	5.743620522564 76	3.495771206106 9	2.964623386237	5.594745129460 1	3.698140539513 47	2.731954690101 9	5.741746331247 46
SCB	8.181463768187509	5.199177892052971	21.382315026081	8.685382355242 89	5.301583164151 7	17.406653645393998	8.610396337941 23	5.235820875327 88	21.316791450214 3
SCC	2.987385706468 74	2.453768968259 97	4.572043234849 68	3.146350336744300 4	2.427825906361 81	4.175830448511 0695	3.048857908178 71	2.386854007255 08	4.572537461692 97
TASCO	20.4728098195709	11.978701105474 7	24.084474134782198	24.59227681123	14.190829073313699	21.790705147723 4	23.235615033090003	14.346281851759299	24.2805966771624
TCAP	3.740478429802619 7	2.429395190714980 2	8.466546851104 88	3.510358120254650 3	2.658816990306 11	7.275217210699821	3.771313364239 79	2.419171362837 98	8.466945401774 66
TMB	3.312408313064896	2.633544477205 13	6.071388801043 5	3.065650164500 16	2.648475226247 91	5.340156928534641	3.424841260729397	2.599938582703 11	6.088472771674 73
TOP	17.32032897176 52	12.578516305852999	26.223461321935	16.822655157692 8	13.045511964619799	24.560338086283 8	17.258071912983	12.617740434061 3	26.101697680576603
TPIPL	3.660145889547720 4	3.855304559417259	9.912145808535 75	3.087864872529 24	3.119786011073 82	9.799719878617 82	3.626077735254 4395	3.651149477784 8297	10.160012805867
TRUE	9.422043763321 06	8.157236547451 829	12.702197652564202	10.528497445609 7	7.344332078536 55	12.004854194 845	9.011058722285 9	7.769545394255 53	12.696913926177
TTW	3.547517010074229 5	2.619258062974 11	5.231589138285 48	3.379090170922 8	2.551097519219 37	4.203160922916 67	3.366372532761 37	2.559355398747297	5.243892089860495
TU	2.607276569572 28	2.405473333558 55	8.036048281842 62	2.564562183586 84	2.397280491334 02	5.952193940631 57	2.600456532707 27	2.394194587400 82	8.036373692257 09
WHA	2.891301713712 25	2.909717034530 26	4.348866530871 52	3.195540439622029 7	2.993334149640 13	4.192998324664 795	3.034536775228 76	2.883022582653 59	4.346757012796 28

Table 38 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	2.855587412744 74	3.213402484049789	3.193073901217 04	2.70173704673 33	2.891371438688 74	2.619077615569 3	2.872496322726 03	3.209693203026 5903	3.194731316010 677
AOT	1.129177521404 5299	1.178866864394 9	1.190777040919 79	1.103013496963 23	1.157140638409 8502	1.131663779907 08	1.128589282907 26	1.175084997843 61	1.191779692306 93
BA	1.634003663900 02	1.416963560739 17	1.640687060157 49	1.452478337296 42	1.224802183806 5801	1.490126862164 05	1.634017691862 64	1.416089222872 998	1.640505374691 03
BANPU	5.043172829492 81	5.219284404361 2	4.546792244116 1	4.696346258560 06	4.658260319090 84	4.583384849222 53	4.926587025485 55	5.181881956151 17	4.540976869777 77
BBL	1.389100823314 61	1.390703905421 31	1.373650115440 67	1.344938703031980 2	1.335883370386 39	1.376821945378 3699	1.403908979035 51	1.391643659092 234	1.373474893335
BCP	1.557207045383 94	1.448942981544 399	1.719132709053 3498	1.500454155028 53	1.420164984933 83	1.534651899587 85	1.557704946010 07	1.451798059033 33	1.719083434726 555
BDMS	1.145538522448 0898	1.188020082283 38	1.156128046581 23	1.106068564281 03	1.100643649920 85	1.123629004987 01	1.144145193539 56	1.188607091185 46	1.155747323703 904
BEC	3.684895890984 4297	3.918035973313 3102	3.082050575442 4703	3.159341086836 47	3.442797410701 84	3.436465175772 3497	3.576905685286 94	3.919723261794 26	3.082726765249 93
BH	1.654474081878 53	1.603588988197 3802	1.715291487425 4899	1.639006303638 02	1.589347088335 28	1.653930536003 1402	1.657826005688 31	1.601704150694 06	1.715017900398 5502
BLA	2.398126761581 72	2.201141817456 91	2.578031827293 63	2.354279595822 46	2.076543817002 17	2.337455632949 59	2.398737524392 46	2.183931376790 05	2.578073301125 03
BTS	1.410518750451 84	1.402526161863 33	1.704450352974 61	1.211750275114 46	1.194313280735 2799	1.221427983800 79	1.398920710476 83	1.401312692689 89	1.707335438542 67
CBG	2.457502346386 24	2.402739129866 74	2.462798421782 79	2.317730317775 6103	2.357655449122 22	2.387538690711 81	2.448956769663 2597	2.392018373269 09	2.465449333324 03
CENTEL	2.349419400981 28	2.343995362984 79	2.397770370493 33	2.27901392456	2.175385824149 07	2.203047570936 84	2.346641491968 98	2.343761293848 8996	2.395528641826 95
CK	1.998971960758 72	1.884700647272 21	1.961195941810 82	1.974428213228	1.786371577822 2	1.918836088414 64	2.022171924009 73	1.890088880847 93	1.960991747560340 2
CPALL	1.498486466735 32	1.449676249457 5801	1.569756166938 17	1.456915559112 7698	1.421173547966 8202	1.485226328996 0799	1.492711814996 5402	1.450612188234 6399	1.569940099857 6102
CPF	2.368938601903 62	2.416765843724 7703	2.786018330726 32	2.335351871003 62	2.324271784337 98	2.368219512126 8497	2.360982103655 46	2.411872008123 0003	2.786172857968 602
CPN	1.355365799927 2	1.364713227835 02	1.611696244333 69	1.297802978329 98	1.291121181086 501	1.333105367789 36	1.358538537755 12	1.366795474755 19	1.611603118275 88
DELTA	2.564989901063 1203	2.603412699125	2.672883061699 5802	2.425300744164 57	2.442338400652 2197	2.659039033441 62	2.578847851924 41	2.610260066338 84	2.675300215549 31
DTAC	3.897600426482 0903	4.202535107592 47	3.964578693172 4403	3.834707187702 9597	4.136164809678 42	3.857268003939 8897	4.000247676283 64	4.205425847 49	3.965701013583 61
EGCO	0.971214484008 538	0.944060408047 479	1.156568656520 9798	0.944594785958 78	0.932624484870 8529	1.005309832232 3	0.972359394707 336	0.944667939763 0621	1.157147966385 23
GLOW	1.438130051647 65	1.382725036587 19	1.481303616338 34	1.389865275968 0699	1.322910368660 71	1.437572360712 68	1.437500280961 92	1.373082216273 4	1.481215417406
GPSC	1.646137611802	1.649100231442 3402	1.649402758431 93	1.706880380069 17	1.687090204444 5801	1.837460006594 85	1.665584556044 58	1.605979905982 51	1.642233468854 1902
HMPRO	1.787319514568 1298	1.763324610649 6199	1.692832061669 04	1.738629732047 75	1.714087687197 3	1.665138603857 9	1.794862341776 21	1.766669146331 11	1.693158774810 8001
INTUCH	1.703020760959 47	1.729399938603 3199	1.844664304877 3002	1.658296033844 265	1.638663962873 22	1.731076383485 1102	1.697015081072 99	1.703536743405 51	1.844881872081 7801
IRPC	1.512984988979 96	1.462666069315 23	1.792989640422 6702	1.436699993718 06	1.440023184046 719	1.547003724642 77	1.503352687947 4899	1.461948096624 46	1.793411538020 32
IVL	2.415189130149 15	2.302383055249 93	2.684703495791 9997	2.335150099404 82	2.209080631743 8103	2.496898820807 89	2.414131736238 1303	2.303380158200 3797	2.685217822357 42
KBANK	2.107364746307 28	2.074747491288 54	2.192828947234 01	2.110521770835 18	1.896240708636 82	2.179485028872 76	2.085381198852 4	2.072210447219 33	2.192911171024489 7
KCE	1.533097689329 65	1.522839537912 398	1.624365183384 83	1.520508245228 11	1.501670665001 5499	1.570163901627 85	1.535261249329 8	1.523650615710 96	1.626051916919 2701
KTB	1.337023906954 22	1.314336790995 85	1.406285790616 12	1.286030603586 35	1.248752986722 6401	1.278210070777 99	1.349803970621 95	1.309690737482 399	1.406437738936 99
LH	1.974124173143 7	1.868108898833 08	1.996720828626 26	1.675050412962 1102	1.623572228950 19	1.688945195140 15	1.954924628716 75	1.862862557438 7998	1.996278091664 5801
MINT	1.778902219764 98	1.882074615915 70799	2.082074091864 716	1.756578274890 05	1.779953883046 46	1.794801833111	1.783794975551 1	1.883981852894 8	2.082127402937 88
MTLS	1.004899585101 8099	0.967295547140 359	0.993533044333 679	1.002607091593 9	0.991808515870 2209	0.971230887939 336	1.011334238943 91	0.959620065086 2591	0.993942255524 211
PS	1.584757490121 2298	1.460549822270 44	1.675490168675 91	1.518017587979 11	1.376759003881 72	1.606929696166 83	1.587482373926 802	1.466139578021 097	1.675557756352 8599
PTT	2.586925409582 24	2.728831187431 68	2.257146392752 7398	2.435205014732 24	2.527635146059 78	2.192412339597 85	2.508159993310 47	2.730537322105 2	2.256811038683 7903
PTTEP	4.353400318586 88	4.725035243423 3	3.922971800578 1304	3.976187956919 9703	4.282051596151 621	3.994848016254 4002	4.191854589787 87	4.677445322374 14	3.919664157124 0196
PTTGC	2.647461875441 34	2.697185995966 47	2.844729412903 7197	2.446037395590 85	2.412491052882 49	2.532796884038 2398	2.576564190029 46	2.685328012627 99	2.846371193300 78
ROBINS	1.618890475349 2401	1.883251153711 6101	1.816867804835 098	1.596103739708 9899	1.740754587540 1598	1.654639524124 96	1.620189632522 2498	1.866305163465 7102	1.818534371747 14
SAWAD	2.085864137134 27	1.789788500957 78	2.019725312502 03	1.977187208073 1698	1.938517703177 57	1.948274640304 45	2.079641612241 8	1.831288077030 54	2.019232393442 0697
SCB	1.851481427361 3198	1.911764618063 4398	1.828084693315 37	1.705708045140 07	1.718849187403 41	1.718850237833 4298	1.817561604559 4898	1.914192587026 17	1.828805067717 23
SCC	1.361740108691 19	1.336056841705 9702	1.363700598913 6202	1.340373314359 87	1.288345723520 83	1.364193004150 2701	1.365656238887 18	1.331848800863 5898	1.363345247322 4699
TASCO	2.415767642374 17	2.410848166705 62	2.512622338871 527	2.441163648660 78	2.412458906338 8697	2.510572693858 38	2.429868183542 73	2.406099314485 42	2.511871242323 0503
TCAP	1.329819555535	1.286812313144 802	1.339978584272 24	1.297233601693 7	1.230619318580 6	1.287623259660 5899	1.334595503514 61	1.289898677551 6799	1.340022726862 9301
TMB	1.756618079252 9098	1.646442206083 85	1.780386748705 3802	1.645942962551 76	1.499108573342 9599	1.675103461722 75	1.751444248773 17	1.639090318786 32	1.782304637032 3898
TOP	2.181926695476 7302	2.326641075805 79	3.310165209537 0797	2.157497684084 1997	2.278908393702 39	2.064323231657 9	2.120254805107 9297	2.345948041520 26	3.309598854800 63
TPIPL	1.890426026277 69	1.876953348965 92	2.169446469724 7398	1.931368651712 1	1.844353795873 94	2.048571634393 6	1.899028968168 698	1.881761312878 73	2.171025308673 97
TRUE	2.793436265283 47	2.689848118502 65	2.755818143713 97	2.688689077458 8603	2.609282976937 9	2.791381086876 43	2.793094426422 62	2.671055829562 39	2.754976600405
TTW	1.019443526320 36	0.974797627336 0911	1.205454479536 14	0.973231041741 33	0.926561069635 0559	1.036645983411 38	1.021057460565 4	0.972359998037 995	1.205572840891 2498
TU	2.556677462067 84	2.137269676302 56	3.246844333455 5903	2.419639819148 29	1.985237643693 7098	2.698349268267 87	2.556903213106 02	2.164976550773 34	3.246780167240 4497
WHA	2.247553160093 95	2.059597670175 88	2.208901433555 06	2.045425882639 54	1.866826343088 88	2.030136034043 0104	2.248891908644 93	2.063798593904 5302	2.211037246162 51

Table 39 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	2.216127864421912	2.38219721284918	2.38848059091457	2.15540693472391	2.23366213810837	2.11732727926285	2.2076161381861596	2.3792793311074	2.3887873894902603
AOT	1.167095580888018	1.194587564080666	1.276649431091153	1.121455190434773	1.190768661304	1.201708106985165	1.164488758158671	1.2019199692384	1.128636037745151
BA	1.35196040663806	1.125038773807177	1.4176106709488001	1.23775976859555	1.12850934304489	1.25388677048618	1.369087773570529	1.235743012395394	1.41772342395394
BANPU	4.860511659594850 5	5.165081300192288	4.48704193087359	4.614271235411070 95	4.60842783238535	4.57539308894817 95	4.920002393556661	5.169602402898319	4.48034738664169
BBL	1.357867621792160 1	1.355284400804 13	1.361311436500521	1.317102856504 76	1.32416877736089	1.316595032513 25	1.369711893024 49	1.355194213013666 98	1.361046859323901
BCP	1.55529474736352	1.447374410147029 8	1.71913279905334 98	1.501927306528329 8	1.42836549709505 01	1.534651899585 785	1.5563442648621 502	1.45126288405256	1.719083437265 55
BDMS	1.1240289144934	1.189732885132 27	1.156128046581 23	1.104272258228 22	1.103503934978 34	1.123629040987 01	1.147741262917 01	1.187677707121 89	1.155785731170 38
BEC	2.891663684244569 8	3.109652874919 27	2.561762352733629 7	2.627053730160 21	2.745681403187 24	2.754569247551 05	2.891579218675 82	3.106036399720 28	2.562064444256 29
BH	2.117631627266 84	1.990376906456560 2	2.21532642622768	2.015629249012669 7	1.920688807637009 9	2.016032197720 59	2.125389637739 18	2.0022175224929	2.215246223951 33
BLA	2.403938005746790 3	2.181625720266 52	2.578031827293 63	2.338827404177 1	2.090151525135729 7	2.337455638294 59	2.401676962174 89	2.201831282246 6	2.577712543382 98
BTS	1.267669763026 17	1.263204883773 14	1.584521462570 09	1.135679231540340 1	1.093447793747 57	1.169979322069 69	1.274399381882 04	1.261572515363 32	1.586342823152 082
CBG	2.46220004633926	2.407397544942 31	2.462798421782 79	2.355102855643 74	2.348029712088 99	2.387538699711 8198	2.442547206847 5202	2.402461551288 9303	2.465449333324 03
CENTEL	2.3318111133401	2.340716148921 94	2.387018584289 35	2.255344778168 1597	2.158534111735 07	2.189309372919 0403	2.358311074924 85	2.345778191198 6	2.385519059326 13
CK	1.735583004251 26	1.756978462648 0498	1.748200037462 0499	1.71258489655 39	1.740257773110 86	1.703482857482 11	1.740866641055 57	1.757413440418 28	1.747228707943 22
CPALL	1.499233728471 71	1.453293681122 47	1.569756616693 817	1.460027527109 12	1.421215253812 83	1.485226328996 0799	1.498296612676 67	1.450934835669 699	1.569940999857 6102
CPF	2.173487586113 96	2.219203367529 2397	2.513602515050 73	2.151052236400 5102	2.171196726785 42	2.199013442384 87	2.168886700380 96	2.218848907022 96	2.512963621720 79
CPN	1.368326297325 83	1.366079410459 21	1.611696243433 69	1.297061852144 87	1.298075764533 23	1.333105367788 36	1.363639374843	1.365700390232 01	1.611603110275 88
DELTA	2.572797351246 86	2.615297739885	2.672883061699 5802	2.444831047678 697	2.461802192351 72	2.659039033441 62	2.575139549035 83	2.603033191340 3497	2.675300215594 31
DTAC	3.624406319614 9297	3.749728437894 84	3.554380722766 0102	3.662239703201 59	3.855751940881 66	3.639398392711 6498	3.637209372300 02	3.747266378838 94	3.557245691024 1397
EGCO	0.971403757043 4391	0.943770325691 1589	1.169501342528 18	0.944739490795 28	0.934137283358 58	1.007962610360 35	0.972403695975 7569	0.945088178732 2249	1.169363644278 1798
GLOW	1.438879481714 04	1.378345323164 09	1.481605342976 72	1.389614352264 2301	1.322007507867 15	1.437557819143 71	1.441450765677 77	1.379622353012 45	1.481724117178 8
GPSC	1.683026696235 9501	1.615990535076 8	1.649402758431 93	1.709205774119 09	1.696717032165 68	1.837460006594 85	1.663558271726 5101	1.665827609526 3598	1.642233468554 1902
HMPRO	1.791855210030 7301	1.753304149632 72	1.675024690847 26	1.729622544458 95	1.694122460385 29	1.652284398741 5998	1.791109103744 2601	1.755083949472 14	1.675387236760 7898
INTUCH	1.703312191635 4799	1.728676721352 65	1.844664304877 3002	1.664761373679 47	1.637504262448 999	1.731076383485 1102	1.697244028013 298	1.701544742530 08	1.844881872081 7801
IRPC	1.516252380489 698	1.492296382878 54	1.975846686820 21	1.441442322559 49	1.417955496301 3499	1.702092294775 43	1.520384374597 4899	1.493846603960 42	1.975384926360 46
IVL	2.33045563732 63	2.221404107711 88	2.785823365908 25	2.260876905670 3197	2.181928833060 86	2.420745972371 88	2.324736267997 26	2.206687420268 09	2.785366748722 44
KBANK	1.721756623441 9799	1.703128530915 27	1.81469717973 97	1.765955070081 9302	1.636549030564 55	1.873919693807 01	1.722906822591 1401	1.712961877152 87	1.815121437864 6
KCE	1.54763416854617	1.532631552765 84	1.652836488332 9708	1.520791209467 26	1.505250708505 45	1.584990593691 5099	1.549892185280 07	1.533059867078 099	1.656127450257 75
KTB	1.350671797314 59	1.313653954144 23	1.406227530173 4701	1.289693548512 34	1.250321508633 801	1.278210070077 799	1.331058838493 49	1.313329080553 85	1.406112892491 3502
LH	1.694956763655 7998	1.619192669703 03	1.765339645776 97	1.623006983276 86	1.516655469714 98	1.652435167856 63	1.692767284871 82	1.618848721981 2802	1.766197208708 0198
MINT	1.789353771987 48	1.910720721606 1	2.030122295387 9	1.731412344466 8101	1.788334772504 2902	1.785291785066 801	1.779169671979 59	1.913206613281 3	2.028660371350 04
MTLS	1.004615614345 9901	0.969345106025 148	0.993533044333 679	0.994383639666 525	0.992087366374 925	0.971230887939 3341	1.005560480983 42	0.968224513171 721	0.993875516572 008
PS	1.545809256334 73	1.451336104884 16	1.584098186311 5202	1.449141097481 78	1.378676512413 37	1.480484737414 06	1.544047004801 36	1.455245193718 66	1.583359904084 06
PTT	2.569421390603 81	2.726121436279 57	2.237396225500 64	2.430236304038 26	2.485587554979 77	2.184001238062 47	2.595150651784 31	2.728965669586 33	2.237046786278 88
PTTEP	2.786522511613 89	3.017941868539 1797	2.783198718152 2	2.744826871721 44	3.013115818274 72	2.850681529778 04	2.782488426874 36	3.019801988310 65	2.782765330492 92
PTTGC	2.638300556523 02	2.696905065629 87	2.844729412903 7197	2.368095776267 41	2.397728785885 28	2.532796884038 2398	2.681577959944 96	2.688392037014 26	2.846371193300 78
ROBINS	1.638630112161 3303	1.879061529306 67	1.816867804683 5098	1.590141587943 78	1.741236174907 299	1.654639524129 46	1.641041744461 4	1.870487225670 39	1.818534371747 13
SAWAD	2.080061110596 64	1.781049600256 58	2.019647783754 698	1.971589856162 58	1.912183898173 1198	1.948274640304 45	2.076317214026 53	1.854772074907 5898	2.019331631163 87
SCB	1.828571837629 98	1.912537255852 98	1.828084693153 7	1.696494179741 07	1.700911407088 9502	1.718850237334 298	1.824794826817 91	1.912720872118 95	1.828805067717 23
SCC	1.365199536670 7101	1.338874219749 35	1.369094415261 62	1.338910953411 87	1.289145571890 08	1.372927241371 97	1.365363626391 66	1.339066005396 83	1.368615735737 22
TASCO	2.921923485010 75	2.941667399644 3	3.187078697151 097	2.9965129924396	3.023145763527 41	3.151121638417 52	2.911727396218 9603	2.915837911368 38	3.186404003344 54
TCAP	1.331111913285 15	1.288704458937 52	1.339978584275 24	1.296932496219 84	1.231681134158 39	1.287623259660 58	1.334553705326 17	1.286010541683 67	1.340022726862 92
TMB	1.751652152139 2598	1.635700351450 3698	1.784435151346 2802	1.641706545163 3899	1.518022490600 48	1.669709100516 04	1.751830996812 0802	1.648193907855 83	1.785512285510 09
TOP	2.184257381407 47	2.328584130495 59	3.310165209537 0797	2.137971664939 69	2.288905377036 36	2.064323231657 9	2.211663243437 8	2.328324105160 77	3.309598854800 63
TPIPL	2.159679272132	2.050645948014 71	2.433616055828 12	2.170134908060 93	1.932807561535 7	2.344854199965 09	2.159035963402 15	2.054252175872 8403	2.434454617217 5097
TRUE	2.323439512796 0097	2.302082210333 73	2.342670582638 96	2.340621637117 49	2.230264567432 67	2.409786734820 61	2.317609143762 82	2.312594518734 07	2.343415268489 55
TTW	1.109219918043 91	1.016664135601 51	1.302355728003 82	1.032841832382 8	0.972482807632 6171	1.092151018708 1	1.108512539906	1.015976700152 23	1.302771457070 12
TU	2.570282543398 02	2.150968337194 87	3.246844333455 74	2.411536277776 56	1.961642953391 5298	2.698349268267 86	2.556193298032 15	2.155167400751 41	3.247074811328 68
WHA	2.254549651114 15	2.063907623642 5202	2.204461216079 1397	2.054105153182 63	1.869778400710 71	2.034799605745 68	2.255354590708 7797	2.078515758397 802	2.206784113717 12

PredictPrice14 : EMD from closing change

Table 40 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	29.509410543388 78	20.304254157884 42	48.623578361030 2	30.331626276946 8	21.091795951206 098	41.00764286643 5	29.727441584613 203	20.441308329021 102	48.6403367172 731
AOT	13.686316044419 7	9.6414255775201	26.399059395771 2	13.552949449148 699	8.8419040130139 8	25.13897437368 11	13.008764669258	9.6506228879368 3	26.339666727918
BA	1.4501667097176 2	1.0356076369606 7	1.7457059984160 899	1.255676317633 17	1.1458856068680 85	1.6087782690302 3	1.3326236991715 11	1.0559646085445 94	1.7452808935252 602
BANPU	37.888977478321 6	22.474961344563 5	95.309247582686 3	39.606447291525 3	23.121939946427 7	80.073882451324 6	36.572094511465 096	22.451927037015 1	95.935146571969
BBL	9.2081215557283 01	5.7139773161014	19.659098140883 7	9.2583767656540 989	5.9243740413517 3	17.041410496678 1	8.8298209864938	5.5974534478063 4	19.661005223153 5
BCP	3.8005896033102 804	3.2692160652219 1	6.1045372343417 4	3.9307811585972 803	3.3586886274698 697	5.8585141987540 394	3.8928256009179 2	3.2022219243754 404	6.0989499454015 2
BDMS	9.2388837917266 41	5.6311586341054 6	27.853381547604 002	9.4999928329170 5	5.6265045963792	21.153165948490 297	9.4861379474256 6	5.8711686617367 6	12.524213115317
BEC	39.045621456226	29.173377166746 7	65.009387220271 3	40.002685540846 2	29.294334573584 2	62.660363582961 494	40.539781801529 5	28.965852806146 096	64.849373683731
BH	3.4018567490676 297	1.9821807017476	12.717726543594 99	3.4750905328681 6	1.9916636835235	9.0603989485599 5	3.3639778017445 1	1.9807646937886 898	12.523091838680 301
BLA	8.5093140070819 1	7.9067735417258 89	29.142312528766 002	8.8721037467652 8	8.3845022450563 6	24.820448358592 5	8.5992554116136 6	7.8030623738754 1	29.220798154406
BTS	2.7403503783635	2.2082947412186 7	7.4716590560289 2	2.5944878555348 4	2.2870248431728 9	5.9228436722687 1	2.5682527901539 203	2.2033029764141 396	7.4734586109488 7
CBG	26.092078712996 503	20.161263722784 5	29.017565115953 197	25.822087016747 503	21.949267104774 1	28.440040183207 2	25.937808331182 9	20.188181867011	29.015297153262 598
CENTEL	7.4257494998521	5.7869237925731 9	25.896862695340 9	7.3748344593227	6.3697034109174595	19.769541724916	7.3239713075411 61	5.7512292080030 3	25.891905634815 2
CK	4.5775570039104 6	3.0757270265939898	17.035058574830 6	4.5697433211329	3.0908118726430502	10.605759694216 5	4.5771100559230 3	3.1417392677282 1	17.408987332498
CPALL	2.8939562205098 8	2.2075578492896 2	8.2988144484892	3.0501390402505	2.2863111212993103	6.1286318715848 1	2.9079553669603 1	2.1999437818426 6	8.3168715973838
CPF	9.3440228733848	7.2949469965872495	20.700741247051	9.8963308314773 3	7.5192559049211 5	19.493532339575	9.3714967543394 5	7.2996785618266	20.723987129038502
CPN	8.7536285988739 6	5.5852308560657 3	16.742520770207 2	8.8164506495074 4	5.793023766978	15.757812260619 4	8.4653138082547 1	5.5597289959057 1	16.7398510056753
DELTA	2.475258473623	2.1999646205845	6.4406724339884 9	2.3818513588581 6	2.2463809361455	5.0925505553856905	2.5706606454877	2.1895292528684397	6.6120898733765 39
DTAC	93.216505586426 28	50.832765818707 2	231.11519975748 7	89.161850587883	49.716031227495	204.28943747668401	91.840584472495	52.045420644572204	230.77772623096303
EGCO	6.8476562323566	5.7165479816972 21	15.048158407471501	6.9164537360238 6	5.7292325818438 8	11.754483170139 6	7.0009682057845	5.7886227728624	15.084079601675299
GLOW	1.7587998902806798	1.5903256572111 8	5.1437590843161 7	1.8442066910333	1.6865146321820 2	3.6663997396426096	1.7286547530385 1	1.5836467243966 3	5.1850103323605 35
GPSC	5.5089323733533595	3.8065460589934297	2.2666849165852 5	3.3209154256022 02	4.7852560499707 07	2.6335560099492	3.9693730476043596	4.0765723938302 4	2.3743190650515102
HMPRO	8.2177441843638 7	6.0058807224124 58	18.123360107945 3	8.5213181368696 59	6.2180984060017	14.841702123851 7	7.9779098914206 6	6.0319364591667 295	18.134623718713 9
INTUCH	24.635881595635 5	17.351974116491	54.302666611071 4	24.912203073666 59	18.146583027924 02	41.887797909644	24.929801925107	17.481204927692502	54.455471913342 38
IRPC	8.1960699441694 9	5.9839909354536	18.707929857143597	8.7689186359185 1	6.0986645040288 2	15.238652694172 8	8.2033511448774 11	6.1833781174684	18.782298888751 6
IVL	7.7800876637691 69	5.9884190599577 7	21.520923418229 1	7.4049606037899895	6.3997428674595405	16.416610255448 3	6.8705726468875 4	5.9830214033315409	21.530895893621
KBANK	5.9259469722914595	4.2686308552811	21.726165771749 24	6.9106128520853 4	4.3800490646211 2	17.235871929569 2	5.9068591624697	4.2184338023250 1	21.701438467 618
KCE	26.709387861505203	16.955901074063 1	57.818075650469105	25.541189306335 8	16.428119902335 6	46.154232226174 6	25.655148099109898	17.004900368222 7	57.881549106784504
KTB	6.0746896253492 7	4.9774919646066 8	12.132720201875 8	6.0724838171608 2	5.0173907140663 8	10.285219293869 81	6.0893233227894	4.9725301570622 795	12.127272907110 9
LH	1.8623729274837302	1.7598533076187999	4.2520443786399795	2.0088290766716	1.7753041237290 3	3.6033098094728	1.8722428605464 9	1.7581005387779898	4.3040443747607 4
MINT	16.231555654864998	10.197176610447	32.510812011075 6	15.450468212752 4	10.589756409969	27.507022450781 9	14.836945213071802	10.437001536727001	32.504049344249
MTLS	3.0318309179246 8	2.6399966208512 3	3.6195866843311	3.0182949870943 4	2.7514350639622003	3.2380266871832 5	3.4050041035011103	2.6633125925613 6	3.6305026827792295
PS	3.4774975223312	2.7850265397605 4	17.968693301948598	3.5088841204517 9	2.8097484357042 5	11.024443620556 8	3.5028083740859404	2.7425626442005298	17.992108977171 7
PTT	14.854950552547 57	11.800661391543 5	22.967573527436798	14.712823181653 9	12.264610715790 5	21.901473636989 5	15.055230935955 6	11.641402731619 6	22.905188983498 6
PTTEP	25.361037226529 7	20.629294094180 5	64.555736562629	24.397894395225 9	19.277767417012 7	58.147677659025	24.751480346572897	20.934161163606 64	64.727052091293 9
PTTGC	4.8727572615599 5	3.7173358298818204	15.967908358719601	5.4217545710750 9	3.6958160184945	10.953233766725 8	5.4400689884771 66	3.8772473806943 5	16.127354368726397
ROBINS	8.2592690607393	6.3895892680475 7	14.373709185710 6	8.4228176050741 6	6.5035348505136 8	12.479044315107 4	8.1962367399205 6	6.2973592660194 1	14.384158732225599
SAWAD	1.9860260746769598	1.7487949130169897	3.0001110032470 2	2.1638141744791	1.8866255527586 2	2.8678323424992302	2.1113690244822 5	1.7500449937730 2	2.9982940585999303
SCB	21.368394157035 1	19.930680771753 1	17.783321411434 1	21.009263561166 2	19.771338718698 6	21.176734981104 8	21.532621420600 4	21.179568959471 9	17.259755103316397
SCC	3.9245047764437 7	3.3894948741301 5	19.868387011059 4	3.9213167370402 8	3.5508689953378 9	10.941018667681	3.9131772226771595	3.3614054825633 1	19.915957784568 6
TASCO	10.639681953977401	12.262938108771 36	36.361338086431 1	14.186090233181 6	9.8910758497385 31	26.843711440234802	9.9341823072225	12.121306787403 36	35.919789225131 6
TCAP	3.7799551643655 6	2.9027846552126 6	8.7563382925895	3.7297026177512 1	2.9252470000418 2	7.4022369413664 4	3.9653999372920303	3.0061194997147 4	8.8362164843065 5
TMB	9.0954286116031	7.6192700053550 5	16.168295595334 1	9.2543157531406 91	7.7643047610663 4	10.196589828309401	9.0800340895167	7.5447137895357 6	16.072370406322 1
TOP	11.908371809445 9	8.3514185386647 99	21.995885034079 9	12.092993181753501	9.4125048583558	23.179584684805 3	11.283420585059 3	8.4117374684590	22.025301391755 7
TPIPL	4.8527545952314 5	4.8230773988582	38.306767551552906	4.8369155481627 5	4.7287292920276 4	32.208579955147 2	4.7329497811109 3	4.7346649425372 5	39.089595727532
TRUE	13.006198998424 1	9.4060947218048	18.622115565175 9	11.922363018473101	9.2373032328340 2	16.557234946589 9	12.474610696394	9.4248539483999	15.644774288751801
TTW	3.5165788893815697	2.7439207797699 4	5.1253679799832 3	3.3473858108357604	2.8255475061524 1	5.3230369118446 3	3.1079466580693 8	2.6827699894550 8	5.1430733614310 69
TU	3.3984457323434 1	2.8544142260771 9	8.2831605638359	3.2074434807575 3	3.0261607383466798	7.6292352227982905	3.3149908049429797	2.8509202205999	8.2834851258334 7
WHA	5.7086556139404605	4.3158228855452 7	18.003644149616 3	5.5116050235955	4.4996544926815	12.589819272294 4	5.7170723141614	4.3904565980919 7	18.366733050418 7

Table 41 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.914925571973309	1.915259136977208	2.668087853858572	1.83169137613855	1.8485798787996301	2.036926363878008	1.848601808919995	1.9129425911470401	2.66766808708634
AOT	1.238028996167663	1.2152222483988	1.285325626177168	1.135932577791102	1.1139687794589	1.1797984260530798	1.2379226292457701	1.219819402092912	1.285408667177568
BA	1.269983414442767	1.049827124115997	1.224768774447746	1.0606269317004802	0.9826304779749819	1.053236020821444	1.271304966668448	1.042751866308806	1.227103909050228
BANPU	2.44966806105338	2.4740101593123	2.7908965152794196	2.390649563001335	2.315251141422661	2.514900007439993	2.4116464577865	2.485039990571131	2.7916830256646
BBL	1.399147555502079	1.3793617938711	1.3780927555989702	1.312412762742411	1.3221041232061999	1.274286637944213	1.382004315864014	1.37917633730008	1.377926255669852
BCP	1.398307925130222	1.411161429005156	1.8636609634298003	1.480793091178116	1.429273990429999	1.510902898847222	1.397937046126506	1.411724175135430	1.863519399669229
BDMS	1.1680361851202798	1.151711509922213	1.153777163529446	1.290141503767854	1.287441899769411	1.159433272725333	1.166202517716390	1.151919146843992	1.16317595585270
BEC	3.6693620059547705	3.6769441861270797	3.147845659999406	3.512273231099786	3.4666021658937596	2.837617686119634	3.657711064021996	3.659244410454005	3.176401038624541
BH	1.556971019199948	1.512923607298474	1.571601183521444	1.546279124241999	1.5229737629104199	1.576647544556570 2	1.558945559942471	1.514075592247265	1.571448667070780 1
BLA	1.947688487562580 2	1.85149260981224	2.143272911955037	1.953152342142820 3	1.82169557904815	1.936851734837 35	1.966234251626738	1.859596769259598	2.141356265678360
BTS	1.121020139709110 2	1.02831453241	1.560231765002 21	1.122078063743840 1	1.03780447658067	1.187346035519 09	1.111013623512 68	1.033295702706599	1.55980217422 02
CBG	1.994768280809398	2.03487914626447	1.993294327936 83	2.038020308349	2.061420166571 53	2.031006498725 98	2.012148542865097	2.033732608142903	1.991707431605 15
CENTEL	1.755455681668209 9	1.709904470474 87	2.145381906060558	1.756791648170 13	1.691270713691 4702	1.857122620073 69	1.752087647004 34	1.710419168307 83	2.143486995813 8902
CK	1.863252740038 49	1.723394967186 99	2.614265785576 67	1.717021323163 66	1.649627590564 41	2.061331041568 65	1.844083455172 21	1.716722449451 698	2.622459535473 33
CPALL	1.505388949675 38	1.503486604784 2699	1.484358019906 952	1.48705469734623	1.476539196263 58	1.442268136216 95	1.500599476731 28	1.502723872731 39	1.483842008092 95
CPF	2.512932761592 69	2.409400922194 31	2.401751690305 63	2.539332805034 61	2.398184431894 57	2.424713054068 04	2.502628966674 26	2.394992172219 2498	2.353128414032 47
CPN	1.266534707755 45	1.231887495614 799	1.357023234080 01	1.320847596085 3001	1.266752583183 1	1.332372291 09	1.272319393961 7201	1.232349857553 96	1.358325258899 27
DELTA	2.119749820137 84	1.864961016119 5498	2.393241861878 56	2.166003239720 06	1.893472423245 45	2.553048976047 4	2.116022372199 66	1.899499006513 77	2.38760495855 44
DTAC	3.926209576528 4903	3.670761211531 8997	4.470583603898 299	3.827168684313 42	3.669412808313 7	4.167912315533 74	3.934635828216 61	3.685282313907 2498	4.473542747549 089
EGCO	0.965798877092 9811	0.952022211132 158	1.18950712909 78	0.985268042154 2159	0.977411773344 9901	1.003007092019 38	0.964943362906 962	0.951767165838 336	1.18956909524 424
GLOW	1.399067155537 88	1.380576605345 3798	1.773630717833 0202	1.461556557703	1.404847479390 14	1.540382574247 63	1.403753792121 19	1.381501457103 65	1.773151864187 73
GPSC	1.572000698160 9999	1.463253982962 27	1.613656112157 38	1.596524462843 8398	1.582772747076 74	1.470589513537 59	1.570180992250 9797	1.461559063519 59	1.610009501773 73
HMPRO	1.596741928522 35	1.579549321787 99	1.772230861341 81	1.516644596904 77	1.517796691848 97	1.565732991488 34	1.596980016498 9	1.563512931596 1098	1.771765750477 38
INTUCH	1.850291659306 302	1.841418323780 809	2.10045738888 95	1.939591131419 6899	1.864486693751 81	1.991007846584 2098	1.850654004070 0101	1.845445955789 5202	2.100453999485 16
IRPC	1.445604951559 68	1.423282643173 89	1.463833853985 45	1.446483515937 3701	1.437513180928 19	1.434982887963 06	1.435545014110 63	1.422941518060 9601	1.464432882511 9501
IVL	2.256866291696 63	2.174300133420 29	3.518316700032 34	2.304687894822 15	2.186598296706 66	2.503684219342 19	2.267218780958 1	2.175296691193 51	3.519280591016 404
KBANK	1.609315860374 68	1.600713370120 6901	1.625664370055 39	1.587250978220 56	1.584254654561 4398	1.510721932076 66	1.612271617777 49	1.598947658748 15	1.626007994584 18
KCE	1.572777296928 79	1.591642260929 78	1.607129400341 01	1.572237482903 25	1.580105745245 04	1.541025557470 4002	1.580607332181 9498	1.592110689530 24	1.606392524498 49
KTB	1.368536172305 39	1.312324875311 91	1.659691787781 0198	1.392515424819 34	1.320293785186 5402	1.552457693955 25	1.366230657270 3902	1.312480261502 15	1.659572521403 44
LH	1.677581860030 19	1.562420611688 51	2.024039722877 95	1.651070958651 09	1.590232084131 63	1.685811786951 06	1.688683807566 37	1.569864637780 1	2.024201250766 78
MINT	1.840919335817 3	1.838564194257 23	2.171403140130 09	1.767722741025 78	1.765770728594 55	1.765561812848 79	1.845059572727 31	1.839587627481 9398	2.171135703561 62
MTLS	1.176875169435 0998	1.150798567753 5602	1.187711293975 52	1.256506191616 2201	1.225359101718 57	1.251114859038 12	1.180138094886 99	1.162188170192 38	1.189249572032 08
PS	1.493309683557 1802	1.413611659202 61	1.923035878645 599	1.395675944093 59	1.351212742804 65	1.474097819553 1998	1.494224583627 32	1.413257170066 038	1.923157623718 9603
PTT	2.010732338785 403	2.06371477044 72	2.304892729238 95	1.948155124673 32	2.021268555524 22	2.028557584549 5702	2.000311228554 85	2.064610929222 346	2.309731977101 19
PTTEP	2.771948602464 61	3.188152049447 48	2.647381150044 86	3.220323212727 297	3.120609855762 57	2.856766514594 72	2.772753218875 57	3.176477096516 92	2.652367551609 85
PTTGC	1.873340263413 99	1.921623632404 898	2.403712190209 07	1.945260564413 0898	1.998288990144 3102	1.987600983665 1598	1.873397904926 54	1.921504708457 5598	2.415504252194 47
ROBINS	1.798777488317 801	1.678977438178 01	2.084704255642 38	1.671184702367	1.594030786646 33	1.727390141563	1.802206896332 14	1.674441521423 5702	2.082927052652 3003
SAWAD	1.625899932422 27	1.539402576065 64	1.579553317002 48	1.581316280476 97	1.566099354464 6298	1.602406543331 48	1.623306769046 33	1.528186493739 19	1.572700173815 17
SCB	1.449974017918 06	1.430109566763 28	1.453069846298 22	1.433085795045 82	1.447158031533 15	1.426058585435 82	1.447190576851 05	1.427887583621	1.453565417874 82
SCC	1.141685028286 25	1.136598151862 92	2.306979939297 3898	1.161350375440 3298	1.150658135545 39	1.206504467716 12	1.145598428675 4	1.136687455602 04	2.305340041040 26
TASCO	2.492305203320 34	2.467844247108 23	2.567318990935 679	2.499186870759 62	2.482095098946 31	2.503562820566 89	2.494391240040 46	2.452445373316 53	2.566611763672 52
TCAP	1.213065585360 61	1.205945626156 042	1.250058836426 42	1.190069956868 23	1.200646856321 8	1.178031905639 2201	1.215682940184 899	1.202485339363 25	1.250139767166 91
TMB	1.429259135912 38	1.465098157849 75	1.866312674157 9699	1.481110025291 71	1.474869449425 42	1.534686610170 2101	1.429732338672 29	1.462598279987 2599	1.874587079452 57
TOP	1.607127752517 902	1.547860486526 36	1.744702952342 42	1.556037789521 901	1.512661336663 63	1.573566275173 6001	1.633781122700 57	1.541690235829 4	1.744410829405 3602
TPIPL	1.849309799168 2201	1.806068376908 66	2.651325984818 01	2.030030653702 95	1.957624370027 5602	2.004241397468 64	1.850314808549 89	1.805451151666 77	2.737942603623 2504
TRUE	2.477889921676 12	2.397632823002 98	2.969495438137 0002	2.502171127645 84	2.543162805107 58	2.544192652120 3198	2.471834352485 82	2.406447242507 46	2.970164066000 85
TTW	0.930270945824 639	0.858560417760 3441	1.437953197569 06	0.899040471008 7839	0.874305110511 6381	1.057088742630 0402	0.925070741126 254	0.857019366373 574	1.438195255308
TU	1.530866423950 65	1.501336584885 35	1.811301329830 71	1.608647694903 9	1.518187734996 7199	1.652276460359 302	1.531257473393 22	1.505291347165 91	1.810987358578 9402
WHA	1.779000303252 97	1.779704857020 55	3.037357807261 83	1.758336392801 1801	1.767218255555 57	1.752830429233 41	1.783922452617 1302	1.782239649857 0099	3.147532747621 597

Table 42 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.795778585898 52	1.801080929689 33	2.209089279417 25	1.782347227149 03	1.778989862533 5098	1.864276087618 0699	1.787387239275 48	1.804297701198 19	2.210125841045 69
AOT	1.362414179860 87	1.313045105052 73	1.461515911772 19	1.312502080544 82	1.226642127270 8499	1.279943978450 7398	1.363903567445 58	1.317506454412 84	1.462090738055 55
BA	1.137908274333 79	0.998537323355 9179	1.165788861370 01	1.037276861584 54	1.003216385065	1.040645937909 32	1.142250275063 36	1.001202806890 15	1.161485222758 84
BANPU	2.444949496636 377	2.467910447879 5297	2.829743243985 17	2.342727669376 4498	2.306055202757 63	2.503743407056 17	2.435742792939 55	2.468898482308 128	2.831015637147 6403
BBL	1.371920243963 13	1.347897092584 15	1.379656574622 551	1.287664511688 89	1.302801273277 499	1.278613596999 88	1.385643054016 63	1.348238629330 79	1.379281330701 652
BCP	1.397522666435 43	1.411400048586 901	1.863660963429 8003	1.477294318817 57	1.428346160343 16	1.510902898847 22	1.395114345708 4199	1.411688455385 4	1.863519399669 2299
BDMS	1.165623874400 31	1.153998385936 11	1.153796595899 63	1.284788220569 499	1.285179713939 98	1.159433272725 33	1.150865541150 602	1.156650318843 78	1.163124087719 93
BEC	3.021801960269 34	2.983392955497 3397	2.634055087151 22	2.761810297390 12	2.829817299718 6297	2.454695055899 9003	2.929875434172 49	2.981433830165 63	2.652367611669 071
BH	1.782916095116 8	1.662792325118 96	1.884545971398 401	1.780167854335 3	1.665796136728 7901	1.811061796484 9098	1.784893231304 7697	1.666090507180 54	1.884386113609 28
BLA	1.958555598341 35	1.860582911470 8	2.143272919550 36	1.959979390048 81	1.822597301986 26	1.936851734837 35	1.965737765405 298	1.862003236490 21	2.141279378924 19
BTS	1.056955070020 59	1.004724412301 0102	1.484749715928 8001	1.056055532922 87	1.013224737505 892	1.073692745972 02	1.056897433932 27	1.004071495479 9401	1.485379521663 44
CBG	2.005073365456	2.032635426954 95	1.993293432793 683	2.044699974205 06	2.061410530697 3802	2.031006498725 99	2.011859684249	2.032427888229 8397	1.991707431605 15
CENTEL	1.759867672675 17	1.717200218890 9398	2.173313601584 27	1.772676975097 26	1.695382922817 62	1.893589284761 098	1.755084122580 7998	1.712480488021 17	2.172463103012 34
CK	1.973928586315 901	1.816390537186 502	3.105245400038 53	2.168856476472 54	1.883748833352 17	2.594177332710 32	1.996905425269 67	1.815454316617 33	3.117456012096 22
CPALL	1.497624275710 84	1.501961908893 21	1.484358019906 952	1.487033643188 37	1.476800976176 08	1.442268136216 95	1.496157013866 21	1.499908353798 4599	1.483842008092 95
CPF	2.259023334359 5703	2.197733639558 04	2.374494552591 07	2.260966147527 698	2.154880880924 41	2.288621177478 98	2.269110776333 58	2.199844058839 4197	2.374853831509 45
CPN	1.267184206994 02	1.232424404293 4	1.357023234080 01	1.322153799615 44	1.266155844140 861	1.332372291 09	1.270183649431 58	1.233636069341 18	1.358325258899 27
DELTA	2.120464806795 997	1.883512953385 75	2.393241861878 56	2.153205381075 67	1.878996790805 4601	2.553048976047 4	2.120671329411 48	1.879709791650 801	2.387604958544
DTAC	3.552883208535	3.376050911819 6302	4.400784747665 63	3.468886245121 33	3.361657621897 29	4.631885306988 6603	3.554626979098 746	3.396106504717 74	4.399927259048 06
EGCO	0.966882285114 944	0.952994203215 739	1.20143124968 61	0.979890603080 4581	0.978037730896 739	1.007049689987 4901	0.965511521729 882	0.952529298855 689	1.201331692932 301
GLOW	1.397085566934 49	1.379423155487 26	1.769589924092 82	1.455127676473	1.396469168707 94	1.537406165576 48	1.397135291234 84	1.380663296701 87	1.768002148591 999
GPSC	1.560288929274 569	1.441194419673 2098	1.613656112157 39	1.612107639354 39	1.579110092391 3899	1.470589513537 58	1.560075103833 57	1.464872784913 05	1.610009501773 73
HMPRO	1.603294572095 0198	1.584998335458 35	1.794094531807 86	1.514888183848 15	1.522538502952 6899	1.562762503769 72	1.598893823375 2599	1.566420154483 58	1.793681298535 8201
INTUCH	1.849194883374 3599	1.846544974071 9502	2.10045738888 95	1.937337957792 74	1.853749022247 5799	1.991007846584 2098	1.848946099971 82	1.846336025671 47	2.100453999485 16
IRPC	1.489788493609 51	1.465734985799 37	1.537237192899 07	1.518277529831 05	1.475398048611 903	1.486135682707 98	1.485451724515 45	1.465511614087 64	1.537871716747 48
IVL	2.272498469575 86	2.166042136246 71	3.649775194267 55	2.240329254936 3	2.191569273756 9397	2.485372693567 28	2.277440038913 76	2.165471773717 397	3.650791372884 7097
KBANK	1.511087100701 31	1.514008033746 71	1.521236493708 12	1.556980665817 758	1.533490770211 29	1.486219447437 16	1.511465101060 26	1.516101000723 35	1.521081058364 9301
KCE	1.591932079502 82	1.604996643566 7098	1.649003797576 91	1.583425727212 0198	1.596872238755 43	1.553906269234 276	1.593068469920 4702	1.605002608302 17	1.648446623424 001
KTB	1.368576179520 07	1.309773866935 78	1.659691787780 42	1.424356730826 36	1.321461358926 85	1.552457693955 25	1.367475038173 299	1.309541773514 099	1.659724005358 29
LH	1.575613576049 93	1.456271363774 54	1.841738149642 8398	1.627663492916 95	1.547658451521 73	1.638492720853 18	1.582442799019 03	1.452026482218 25	1.842637553313 589
MINT	1.897386299256 85	1.925654230024 19	2.678883581180 78	1.847479307218 302	1.845603175717 06	1.873783235361 86	1.890581194671 09	1.913284825280 68	2.675915541771 6203
MTLS	1.196844072903 17	1.164269792548 2601	1.187711293975 52	1.253133666977 47202	1.179644815501 52	1.251114859038 12	1.196241192557 6	1.154877208997 4801	1.189297070900 899
PS	1.628523107175 24	1.532802284380 901	3.046304903930 6798	1.739150419393 36	1.548609668625 48	1.964677280873	1.629095067709 8799	1.532402849691 59	3.070430743171 32
PTT	2.006827079122 35	2.058613538381 08	2.298977657799 29	1.946803115576 51	2.016046910088 7	2.026762482192 61	2.008409398457 25	2.055988887000 61	2.300635648374 61
PTTEP	2.400336214510 31	2.553371994528 92	2.392057990833 41	2.440849491858 87	2.503425144140 139	2.352832966432 87	2.405601232688 74	2.552911383065 07	2.391649127033 45
PTTGC	1.870716495283 64	1.921776675232 851	2.403712190209 07	1.941851721031 1899	1.990765927521 3998	1.987600983665 1598	1.870677490305 18	1.921467584449 99	2.415504252194 47
ROBINS	1.798578968340 82	1.678590619170 3701	2.084704255642 38	1.670235143270 13	1.586624509394 129	1.727390141563	1.799888679584 01	1.668746221487 099	2.082927052652 3003
SAWAD	1.628953345386 18	1.535564123186 58	1.579553317002 48	1.583425280951 26	1.558029867132 93	1.602406543331 48	1.623613803735 9998	1.526584509006 32	1.572631321127 89
SCB	1.450033281791 86	1.429837857812 93	1.453069846298 22	1.431849811209 64	1.446030442411 59	1.426058585435 82	1.447318891119 96	1.426861605952 33	1.453565417874 82
SCC	1.143864278890 1399	1.137412631929 14	2.380435361531 505	1.183863981096 59	1.153211108838 17	1.216205850797 12	1.146544217840 899	1.137418079164 89	2.381586663039 5903
TASCO	2.781027217809 1298	2.662274297598 28	3.124966993120 31	2.820031316339 8697	2.740205282704 86	2.891428739509 68	2.781576972581 39	2.660217810425 45	3.129392767695 5997
TCAP	1.216274837238 45	1.205853880896 9101	1.250058836426 42	1.186960040666 51	1.200674504773 27	1.178031905639 2201	1.215806776183 66	1.202439168475 699	1.250139767166 91
TMB	1.426509699468 12	1.465389882720 01	1.853429015096 56	1.479390169612 06	1.475821749827 71	1.547383563927 25	1.426769896241 19	1.463498055105 55	1.866870663022
TOP	1.632389752387 64	1.547403790088 58	1.744702952342 42	1.562575570510 27	1.513669164210 09	1.573566275173 6001	1.646471677874 799	1.544334628596 84	1.744410829405 3702
TPIPL	1.915647367932 1901	1.865963909057 965	3.072790407138 78	2.221005326489 63	2.163662929743 08	2.198606828669 81	1.933655182669 9502	1.867471125867 7	3.141294526495 58
TRUE	2.182748606051 96	2.193510492210 593	2.533050639593 34	2.348191587857 63	2.287424135856 7597	2.317257265035 22	2.181777576200 92	2.194300790524 69	2.533257849041 05
TTW	0.967161166721 3561	0.889064520537 4911	1.547000135923 61	0.920658784858 4941	0.874432538351 096	1.081298747872 355	0.965435714111 535	0.881978055053 758	1.54723656653 37
TU	1.531697072353 29	1.504211423465 85	1.811207547160 9	1.604761024113	1.524832089051 8399	1.652276460935 92	1.532044733893 96	1.504260199485 394	1.810957759045 74
WHA	1.767651786905 7302	1.779426091324 96	3.104212133728 46	1.751308177038 84	1.764942654885 47	1.783610319771 4	1.770284664660 22	1.770855552477 4399	3.208176038726 103

PredictPrice15 : EMD from tick change

Table 43 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	28.0283227246846	20.8184407438074	50.6058134556704095	28.9789925708859698	21.837979384144298	42.9369882093312	27.693554675121803	21.26379000772349	51.0050312182148
AOT	19.2446875420612	12.1607109495198	46.4589835300077	19.1636839118047	12.0408992901442	41.5150448546	17.6160470199259	12.258059467956201	46.444118171536
BA	1.273245111823575	0.9868915074536 78	1.542668676431 09	1.147391176712 41	1.119544212325 76	1.275903665984 21	1.301425027356 17	0.991055900837575	1.555001917297 88
BANPU	27.0666394493137	14.4597653413138 99	96.0936663685032	29.3374642772548	16.1951522386507	75.7901564847577	27.7576389816689 97	14.922504074745198	96.1679517517802
BBL	10.6075706235208	7.0279663693808905	18.747712341887702	10.4078933238187	7.2037378684960 01	16.9313766923573	10.8632905668564	7.0856058380758205	18.873226839418102
BCP	3.8873675379017705	3.2556038095499 3	6.104537234341 74	4.063947092179 03	3.3636484567718 7	5.858514198754039 4	3.7337204723465898	3.1901452397902	6.0989499454015 2
BDMS	8.373129399571 6	5.3017260114783 2	23.833574409155 5	8.389146524572649	5.2876958463935 9	18.850482883065 5	8.962131564077 57	5.376494947989889	11.691287879215 8
BEC	41.601760113588696	29.480606903076	67.5466276642857	41.8033906758817	29.6463400121299	64.8920857301929	40.9993926546061	29.2266740161521	67.3932694734089
BH	4.802725395536 51	3.2159022712662 6	17.213990701776 4	4.986854386607 23	3.0058188003120 7	12.896639214178	5.0324761938936	3.184297794588204	17.181515895901
BLA	7.762962565487 66	7.043223161613341	22.763892453895902	8.40833069520958	7.4255711437483	19.300368716480 6	7.962382931378 85	7.05485629370884	22.72451930161 14
BTS	1.668431180589 12	1.668050523396 18	3.495239441442 74	1.725158005939 01	1.8422696125541802	2.657511795433 36	1.590329184358460 2	1.657405721856 96	3.495878045444 23
CBG	26.2642740859121	19.8405700356383	29.017565115953197	26.0570859688075	21.9715322578975	28.440040183202 72	25.940453610450 2	20.21658357617 31	29.015297153262598
CENTEL	12.345616976305001	8.65275416012045	27.8104092049505	12.3623212654929	9.32352299390111	24.673314625683602	12.2811659949034	8.679810348398894	27.97934876566044
CK	3.18049237138036	3.1043177795219297	17.7005854710954	3.3593931942280997	3.2405265150579803	11.445730020644 6	3.231984044426 67	3.09236235711401	17.671274314678 74
CPALL	2.97876376020829	2.2014082061982	8.2988444484892	3.1618741384894102	2.2848692955758 5	6.1286318758481	2.922163807003 82	2.198429074448 17	8.3168715973838
CPF	12.6222309079837	9.4121390356334	25.5197965612687 98	12.3290310535144	9.4600740935179	20.160111329321 7	12.655776197873 95	9.67774540812235	25.652080069104098
CPN	9.1236652391645	5.5691047311922 71	16.7425207702782	8.8017758568155	5.9013792207641895	15.757812260619 4	8.8080874133920 1	5.582966592365	16.739851056753
DELTA	2.596633891549 27	2.1981833807904	6.44067243398849	2.393180639734 11	2.2427372213914 5	5.092550555385690 5	2.572773675376 08	2.191548724417 05	6.612098373765 39
DTAC	125.013159740485	99.5521880858746	206.2885127349720 2	124.745522770429	101.48473267657201	168.132621489279 02	125.902534011543 01	109.317254149976	208.081746128544
EGCO	7.209946929931429	5.7905720660205 9	15.013154863893	7.006286464201 81	5.81953009310509	11.773122650293299	7.144432605343 91	6.078772721599321	15.050465632468802
GLOW	1.990155232893180 2	1.8112987308495	6.184198892190 17	2.1171762572929	1.9000484793577398	4.092386726911 32	2.005083277754 73	1.792440557843 13	6.247376105627 49
GPSC	4.046394516717 27	3.8411342369420303	2.266684916585 25	4.268003542328 32	4.720777995695 21	2.633556006994 92	5.007863430971 14	3.986033471169 09	2.374319065051502
HMPRO	11.00349352261 79	9.9683898740944 4	17.999503348929 3	10.991451394410701	10.0605780289012	14.395699477499699	10.967385429895	9.918296363321689	18.029987392491 5
INTUCH	24.444006377605902	17.366211116467 78	54.3026666110714	24.120054898967698	18.13378429845	41.887797909644	24.150798939099198	17.38338777421 69	54.455479113342 38
IRPC	9.712553635809309	6.7227967207060 2	23.465144073814	10.0169584944272	6.8927271620819 2	15.493260063472 8	9.3688960653157	7.212029678488011	23.466198130361
IVL	8.970273183649 83	6.3228675170459105	24.7256642887472	8.7548575893912 8	6.7794351968082 1	17.895673758439 9	8.759825520734 78	6.372624959737 23	24.73071548334 86
KBANK	8.77935182758601	5.6234010226114 4	17.857067525154697	8.681256341831 08	5.6959604821741 5	15.820358303510 5	9.098108468860 7	6.188394162292489	26.057300955250098
KCE	34.916823755737 4	24.767390430659 7	62.216886133174 1	36.415235886906	24.023079321885398	55.881017517622	36.142843297 34	23.773913093908 01	61.876677623370504
KTB	2.620600866855604	2.1937862300696 3	8.748638933937 37	2.836779654967 76	2.218054925070 22	6.742670339896669	2.650457638625 86	2.195212944937 54	8.733482769125791
LH	2.690354953437 64	2.2796965414681	5.341252826506699	2.602340123462 44	2.23525922767 27	4.315857181855 48	2.666730759122 44	2.328375703541 49	5.361963138821 99
MINT	21.496712090889 41	16.599712040123 9	35.47913901810 26	22.447801022574602	17.07432060022 62	33.465814214749 9	21.746838089035 8	15.819856690754 3	35.41461166976 43
MTLS	3.396252323626189 95	2.6420918656847	3.617076315991 15	3.108513676928 22	2.842515537787 2	3.234036760949 93	3.388438914710060 4	2.644677967890 82	3.628110755762640 4
PS	3.625345066196 32	3.2275228442097204	36.6285822700677	3.619293184433996	3.355144842135 44	17.230966487874 2	3.693831004726 68	3.244332975159569 6	36.59137421690 84
PTT	14.757931795937301	11.520366905665	23.218133369030898	14.33803957358 07	11.884467167569 1	22.089920015406 7	14.340893989205899	11.253979691 1369	23.122344952463 7
PTTEP	57.564615322060995	41.69693188166 78	86.9265599409958	56.824357068958	42.174771307716	87.026536143458	58.509190255024 1	46.277141714750904	87.2367705532665
PTTGC	4.982420193012 83	3.7587496171965005	15.967908358719601	5.18253914645529	3.737374679829 19	10.953233766725 8	5.213291764948289 5	3.929356980916 09	16.127354368726397
ROBINS	7.994719460589 88	6.3987502079899 8	14.3737091857106	8.182249903414581	6.4747023528165 21	12.479044315107 4	8.317397828131 97	6.402598356173 42	14.384158732225599
SAWAD	2.154605678601 77	1.8826389061981298	2.899150169832 08	2.244026789168 3103	1.996666566013 86	2.625978175606 24	2.104741677315 39	1.900180728058 4802	2.898779067132 03
SCB	21.378343107142 7	19.929241493199 7	17.783321411434 1	21.078208241450 9	19.771945753994 8	21.176734981104 8	21.594748185303 3	21.190058474409	17.259755103316397
SCC	2.323441869037 57	1.8390926836672	4.676949580378 9	2.353144181676 72	1.8570189297817001	4.081652429470369 5	2.426757792044 34	1.828915676415 05	4.683021813852109
TASCO	12.657849134208 1	8.977768868149 71	58.024818094865 5	14.819097600224099	8.230848432366509	38.483287558435194	16.189639075452	8.987229468320109	58.245793363452606
TCAP	3.744738492628269 6	2.9033845071426	8.756338292958 95	3.7063412633228	2.960745941873 75	7.402236941366 44	3.739224043516 45	3.004153817533879 8	8.836216484306 55
TMB	2.958150731752 37	2.4050938754967 9	5.757752134861009 5	2.921506562644898	2.426150845759 3	5.208520024287281	2.819611234387110 3	2.396302693330 73	5.766121605449 23
TOP	11.341791136643401	8.397931707159 78	21.995885934079 9	11.208566180482 3	9.389213372501 27	23.179584684805 3	11.786395898100 99	8.468767471553 84	22.025303139175 57
TPIPL	2.886548746654 41	2.597134733835 97	10.642396079667 8	2.820983477108639 7	2.754100681839607 3	8.125508090070 7	2.994991090708 35	2.559401172167629 7	10.698661014978 2
TRUE	16.655503637196	11.998004827346 099	24.9571856824448	17.052823945947 2	11.916115587145901	25.421358892839 4	17.015929363627 098	11.995021536141	30.951640031162604
TTW	2.219318730519 07	1.8340571637121599	8.042077957955 75	2.113990661480 81	1.917329547418 29	4.618583254288 98	2.145100104730 53	1.798368971240850 2	8.044429941551 599
TU	3.1182680767475	2.6303073281 3	8.658624330004 509	3.038714082349 1	2.792058458897600 3	6.096463693140 739	2.941108353150 8	2.602833748925 98	8.658289032210 27
WHA	3.674486949348 31	2.759627370054 04	11.005792971268 9	3.686794796379 62	2.796619207128 41	8.191920203089 01	3.692165185469 73	2.779058810873 13	11.216452749150 2

Table 44 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	2.039900888634 49	2.156515171822 85	2.492476826999 38	2.008180670612 31	2.105439421438 16	2.064475392874 36	2.020157193425 56	2.166079943703 77	2.495428980359 62
AOT	1.263702823672819 8	1.251250577378 4601	1.278463402089 6	1.196119473616 53	1.195826091424 04	1.152837863990 19	1.237239133273 9	1.261128952445 28	1.278567021293 67
BA	1.008456693293 82	0.945239759475 194	1.106787920669 68	1.105796495750 7	1.048610546091 92	1.120375422211 6601	0.999732328443 474	0.948838353412 108	1.107220190113 78
BANPU	2.663044016198 3597	2.805918203954 71	2.768293832970 593	2.619850866364 16	2.634410540265 5598	2.479620933031 62	2.675417775652 73	2.820382469748 88	2.765495285766 44
BBL	1.334262015462 3401	1.343430876642 3	1.438189170080 4301	1.336887886106 03	1.313483659746 9999	1.302961634137 2598	1.326935967275 42	1.341585189235 172	1.438265976126 0101
BCP	1.39641413388 994	1.411035175002 63	1.863660096342 98003	1.477920631654 9398	1.432405934081 49	1.510902898847 22	1.394236263701 4001	1.411832997894 00	1.863519399669 2299
BDMS	1.222131729374 61	1.207246878501 06	1.338558887474 11	1.401857040958 88	1.304714335902 4	1.209433518287	1.215178247697 99	1.207568838188 8499	1.337373240333 816
BEC	3.247093257311 7	3.318920468635 203	2.968461492431 19	3.232033242688 74	3.202650022448 5797	2.615494766920 4303	3.280891975946 8202	3.324283597484 56	2.995575742399 29
BH	1.457573345569 42	1.437059469258 37	1.522811726797 96	1.484879913349 97	1.483943403625 64	1.477636923977 196	1.462318615600 07	1.436742201975 06	1.522486149135 42
BLA	2.209534242742 59	1.931207066269 72	2.314159203862 76	2.086120703645 1	1.869463844536 65	2.093250004687 11	2.209722004466 93	1.944097614826 2699	2.313896986922 528
BTS	1.064384918634 83	1.027473472857 34	1.582259244782 85	1.164203610087 17	1.072906550938 52	1.166356490966 2198	1.059306609094 089	1.028899919197 3	1.581959043977 09
CBG	2.004722113169 06	2.028654904568 7	1.993294327936 83	2.032811173931 7097	2.060822549958 91	2.031006498725 98	2.004923734826 02	2.032020499510 64	1.991707431601 55
CENTEL	1.856321422657199 8	1.873333606410 56	2.416922724669 3104	1.688409955048 87	1.792533764923 4399	2.111288695491 38	1.838039285934 8602	1.871605386727 06	2.416380131505 21
CK	1.730181021871 64	1.607116224155 36	2.634598593868 75	1.860658718341 12	1.756410970022	2.074896491680 83	1.696087034368 5999	1.623885533651 744	2.637093333934 5997
CPALL	1.508168176905 32	1.502091363922 06	1.484358019960 52	1.486888445662 06	1.476320629631 04	1.442261302169 5	1.508305056364 14	1.500628347935 7099	1.483842008092 95
CPF	2.25279145631472	2.1925746161439	3.155897440008349 7	2.200794318088 5703	2.172512761554 95	2.181076613216 7902	2.245252141480 3703	2.188532166811 95	3.164453851205 2695
CPN	1.272204729994 07	1.231522075066 36	1.357023234080 01	1.322974312587 6	1.267235360877 74	1.332372291 9	1.266478456824 87	1.237546646980 6	1.358325258899 27
DELTA	2.122183942723 89	1.903659698251 6099	2.393241861878 56	2.157269972869 25	1.892951742023 56	2.553048976407 4	2.116764316668 57	1.876915680147 73	2.387604958554 4
DTAC	3.493641257611 29	3.624970068400 39	3.895896657891 1403	3.314931064371 54	3.342986210318 05	3.244412396699 84	3.450755944677 5	3.411619273849 2597	3.896189096855 2897
EGCO	0.960152964317 51	0.948244067295 342	1.18741743700 24	0.976924437546 9209	0.977083222178 749	1.000085064123	0.959977782055 733	0.948239818455 1729	1.187582656638 5699
GLOW	1.388189540527 12	1.375573970758 286	1.765780403288 552	1.462043714206 92	1.402355787754	1.513907726302 84	1.398774582902 57	1.374525700300 09	1.764209330222
GPSC	1.559563331741 41	1.464969981650 699	1.613656112157 38	1.656839086343 2698	1.580804036778 102	1.470589513537 59	1.561057950257 5699	1.447850896298 0898	1.610009501773 73
HMPRO	1.601630742056 94	1.629945149250 76	1.524898241145 75	1.502141415566 7	1.522224041418 5901	1.487216679961 9602	1.588586968362 9001	1.563329707833 0701	1.526397955168 201
INTUCH	1.85145000044631	1.842742020060 7998	2.190047538889 5	1.934150542417 91	1.866198471046 1	1.991007846584 2098	1.852343825553 64	1.837369763159 4001	2.100453999485 16
IRPC	1.322035442978 2402	1.323155666080 39	1.692096057891 1403	1.360387444679 67	1.331748752196 5998	1.343223836655 4	1.323996148543 1802	1.322471210111 51	1.718391858577 74
IVL	2.180291884537 96	2.155101096895 66	3.573261777195 13	2.234516691453 8	2.144120284021 9	2.501052418355 82	2.180813685695 4303	2.157951112222 83	3.573865777075 496
KBANK	1.470360652607 23	1.551732044334 99	1.952163770479 22	1.470829686065 29	1.499575619076 3199	1.458159556860 31	1.496556223524 3302	1.552143030088 98	1.853727660101 97
KCE	1.496846257386 49	1.512568749624 5301	2.110303459797 45	1.486171520906 45002	1.507047773378 64	1.572441487238	1.501433348499 8401	1.512868912026 6299	2.109940488577 092
KTB	1.377077556094 71	1.286040853165 2001	1.559679279785 66	1.356136637306 05	1.311235168862 43	1.483730663218 8	1.379698501814 8	1.289715571483 94	1.560543571589 99
LH	1.643442890013 001	1.564686763654 86	1.924343010850 19	1.588753687883	1.556542522928 88	1.679066436523 1502	1.640111139060 205	1.561822389142 4299	1.928093427625 701
MINT	2.058013874968 72	2.096374556350 96	1.926786324990 1498	1.849670786219 5798	1.929932585826 968	1.664294956056 03	2.074119827226 898	2.056756673227 43	1.930809594679 22
MTLS	1.176885787548 5599	1.152660761690 37	1.188185641520 7601	1.244918954344 13	1.212334866244 74	1.250609395754 05	1.178166345397 15	1.151672555191 352	1.190177982311 46
PS	1.397842403281 63	1.363289529806 82	2.111948773875 62	1.353623686790 04	1.370881914787 5	1.545851119156 6902	1.396633701883	1.349439553366 862	2.128232466136 8204
PTT	1.993313510830 7298	2.048458558919 87	2.301296842837 76	1.939727334211 3701	2.013467746815 28	1.999230896966 558	1.953446373874 2398	2.049665437469 6	2.302382421866 31
PTTEP	2.423127739994 51	2.470982661301 47	2.674231960120 62	2.448743818989 58	2.389012516497 5897	2.411356066671 2797	2.412841358999 3397	2.459661414152 22	2.674852596710 397
PTTGC	1.872774718994 78	1.923660859697 3	2.403712190209 07	1.943730411719 17	1.999304700577 4403	1.987600983665 1598	1.873704120728 42	1.922571805420 6	2.415504252194 47
ROBINS	1.797574015524 7701	1.668831387099 05	2.084704255642 38	1.671198643434 16	1.589512911166 7499	1.727390141561 3	1.797823094701 5301	1.676755969647 819	2.082927052652 0003
SAWAD	1.592581091772 91	1.493562935451 82	1.586769735290 9902	1.568733545209 6898	1.529395296702 4098	1.579779258966 24	1.584048624247 94	1.504153486941 1601	1.584840424916 6698
SCB	1.449634291462 43	1.430301560904 08	1.453069846298 22	1.432440801047 23	1.445643770460 66	1.426058583453 82	1.447464792316 95	1.426149047271 69	1.453565417874 82
SCC	1.308492960292 54	1.266393867219 9	1.246338628999 93	1.252649710989 78	1.217647277735 49	1.303683708464 43	1.305450383388 95	1.264636854092 62	1.246237932454 45
TASCO	2.393563312886 18	2.391817028439 7	2.726870551222 12	2.509880213379 5397	2.474147220008 94	2.516943529874 5197	2.395981614907 74	2.381031248104 25	2.733875671580 41
TCAP	2.121682033622 78	1.206068974683 52	1.250058836246 42	1.191256458621 13	1.207509418450 96	1.178031905639 2201	1.213388419295 6699	1.202357820674 51	1.250139761766 91
TMB	1.841425134995 1598	1.786997119040 4998	1.850927203685 702	1.818908644860 1601	1.747870823375 045	1.893874984352 53	1.841591345696 36	1.766244597596 464	1.849584334071 598
TOP	1.632174867146 45	1.547211794844 84	1.744702952342 42	1.561314926664 45	1.510890688976 65	1.573566275173 6001	1.643623643714 55	1.549909911912 96	1.744410829405 3602
TPIPL	1.952552101787 4699	1.877649466074 92	2.336229176393 28	2.296104675321 43	2.046301039336 28	2.449539443764 28	1.953939867491 6002	1.872927725026 19	2.325368791294 5398
TRUE	2.435735954595 902	2.365749778837 01	3.253728098130 89	2.697604485568 78	2.746235773522 88	2.705042722451 78	2.440729985523 95	2.376521792021 86	3.270865025862 68
TTW	0.92005314467 86	0.891060459832 1169	1.109390325617 65	0.981438673091 2981	0.950232465300 626	0.952154924947 51	0.919234565164 433	0.892712830191 548	1.109720679074 66
TU	1.581528884962 5901	1.535754453408 313	2.219493070139 55	1.710228247432 45	1.557631674048 26	1.729702421328 65	1.580492713756 54	1.537375095361 43	2.219864703749 803
WHA	1.895842639532 3202	2.089863228795 353	2.515632112061 99	2.000536701772 1	2.054419266523 64	1.901883010546 65	1.896136889000 15	2.089836194345 66	2.516483477713 46

Table 45 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.8464329766510201	1.8897762077224298	2.004439851143226	1.8274114763033498	1.8848615967501599	1.841870666516866	1.845507005445306	1.89143332915231	2.0043321744714198
AOT	1.405642008870156	1.387274337691577	1.478458035597708	1.45097490228708	1.3936874459362	1.351930771120624	1.418577752794291	1.390988011381624	1.477109055197681
BA	0.9892050001222873	0.9392321152516723	1.084151212120748	1.035813872920285	0.9824473992141941	1.047457630120748	0.9770489705229901	0.9293990682343363	1.0818879502299201
BANPU	2.735218244785603	2.8197227678604198	2.816701561397	2.6464676182480398	2.619936707872114	2.478134242898895	2.745088142366944	2.82588230550958	2.820947784942918
BBL	1.3344279911886499	1.335059592027981	1.543684208036752	1.2828585305331102	1.2977359148733	1.269667948997833	1.332371004847377	1.332409032063331	1.543650417619461
BCP	1.396571998557592	1.410869994037133	1.863660963429808003	1.4778975624911201	1.4320336617792502	1.510902898847222	1.397446034330075	1.411456069848666	1.8635193996692299
BDMS	1.2142613714825798	1.2074469715340401	1.338555888747411999	1.428242341540671	1.2966262098702002	1.209433518287	1.217268727217669	1.209177133592741	1.337499194068571
BEC	2.7756202720794096	2.794960074083223	2.558571884057162	2.756033460128962	2.725065109797087	2.36994189262266	2.799704468523666	2.7997681264509002	2.576918730283777
BH	1.5582668143232101	1.49710597693825	1.566724654341957	1.572983063526676	1.54914515949839	1.540765395973276	1.559161264260039	1.49681255355241	1.566767579182941
BLA	2.215102983536877	1.9373402429751398	2.314159203862763	2.086299369750474	1.8609174623854599	2.093250004687116	2.21845766302149	1.9408125259209499	2.314028696506545
BTS	1.157390055580061	1.049688090634199	1.539932059902399	1.094123341818741	1.030618107129454	1.145537541058471	1.157204592620454	1.0483116835203	1.539600185894191
CBG	1.999680732633991	2.033249433388905	1.993293432793683	2.056904064459151	2.055691685846972	2.031006498725991	2.017187493772082	2.027731639191	1.991707431601552
CENTEL	1.845564877852857	1.877488608130102	2.445101962301177	1.730676740396409	1.822538128674521	2.115054921525209	1.844781898956	1.881025370707862	2.445608371602741
CK	2.12751327499128	1.89541668826459	2.826750315520960	2.114117609596903	1.9191516328517	2.175305985577939	2.097503766672381	1.897727570025003	2.823958164393838
CPALL	1.501586390600821	1.503021516563381	1.484358019966952	1.486966620910921	1.47793545900812	1.442268136216951	1.506667387806301	1.502184010496721	1.483842008092951
CPF	2.186034546085671	2.178557647255140	2.304099554499	2.149941247155053	2.13991892697392	2.151656625942070	3	2.185520483234119	2.170477517131599	3.058451276111089
CPN	1.266633600861830	1.232972763003340	1.357023234080001	1.322328910581289	1.2660450461008	1.33237229109	1.267007451955560	1.239176679347749	1.358325258899270
DELTA	2.117978346593311	1.893969042152250	2.393241861878560	2.153243611857140	1.8920041717643599	2.55304897640740	2.114681209406960	1.893187388558508	2.387604985544
DTAC	3.30709885234750	3.383581461014250	3.717912551053190	3.17538564660010	3.20770621613912	3.253964658902103	3.289186857207	3.273420303440338	3.71742647353410
EGCO	0.9588899708725731	0.9471582312625321	1.200471621126770	0.974523415021699	0.9750603763062	1.000070971854610	0.959889682612308	0.947002256948361	1.200485358120010
GLOW	1.384227670054980	1.373554818944650	1.775719680638402	1.456033936365662	1.404216692505450	1.508008259719680	1.388968351166	1.373634410115600	1.775581001273765
GPSC	1.576201072576660	1.46000033497483	1.613656112157390	1.692665400491702	1.593893153510070	1.47058951353758	1.572843185336660	1.460046903097900	1.610009501773730
HMPRO	1.603492481917590	1.605133394125930	1.520918076884830	1.492935159363980	1.517894512449290	1.481549696441969	1.570618147284570	1.556043314757502	1.522407622069850
INTUCH	1.854410483954199	1.836181646366329	2.100457388889500	1.935546433550025	1.86794953295732	1.991007846584290	1.844162818576410	1.841070696013420	2.100453999485160
IRPC	1.33116720885573	1.343616575548435	1.821338957372606	1.388710852867560	1.356344497005670 2	1.361299265593101	1.337748498912530	1.343464350067080	1.850780821257510
IVL	2.239338842003180	2.176895830367050	3.659521392597720	2.269792183359380	2.203928642787895	2.710647797427263	2.239321217656960	2.179506849737680	3.661885733489330
KBANK	1.426213871754264	1.507211114870390	1.710855737388200	1.436808944646510	1.475260558189140	1.488966243390150	1.429957089150410	1.506476666114160	1.715976344883040
KCE	1.508785901724	1.519014319941590	2.132688284247560	1.494561293980398	1.520926956442	1.599565221916960 8	1.503427275067430	1.518964052254390	2.132817372420280
KTB	1.382536439351090 2	1.296163252111700	1.559679279785940	1.343560058020850	1.31698615364986	1.483730606322030 1	1.369766365072050	1.286848903114960 1	1.560334137321260
LH	1.532040057580851	1.483136280421139	1.750573654863110	1.620754627346010	1.540977876068	1.689508978959290	1.536967562805160	1.486256902145730 1	1.750960654162230
MINT	2.500994110409530	2.529729288114270	1.967754253512210	2.085610844502730	2.317499399147760	1.774940049385720	2.402486960705940	2.473122645174940	1.968641690842469
MTLS	1.174899988133180	1.160081478643330	1.188185641520750	1.253482740925110	1.190401856702610	1.250609395750459	1.177066256037440	1.156664860323790	1.190097317550120
PS	1.639518151579840	1.561667698922650 1	2.257612662874060	1.565970744423490	1.636834385172780	1.910767563802680	1.590955969118550 2	1.538289034954089	2.293146855526330
PTT	1.973175528504630	2.045025349355060 3	2.289528939989540	1.939371124306399	2.010269858725830	1.991946164760150	1.983462061990300	2.004368004502970	2.290345379556820
PTTEP	2.273707102045650	2.291922510455820	2.418013148952840	2.296094220498390	2.253668270589110	2.273020482570820	2.264179775357	2.287448246379780	2.419087020246275
PTTGC	1.871651929652890	1.923483050180430	2.403712190209070	1.952493153383420 2	1.99323143797961	1.987600983665159 8	1.871495826266150	1.921649265881680 2	2.415504252194470
ROBINS	1.799500475525	1.671723297584900 1	2.084704255642380	1.663378108493860 1	1.596294051930990	1.727390141561300	1.800576492327509 8	1.682225201897398	2.082927052652000 3
SAWAD	1.597509720543970 1	1.492339906831179	1.586769732599080 8	1.567009783175680	1.532226431070360	1.579779259866240	1.593939130364630	1.499624112567230	1.584568740523740
SCB	1.445830742729290	1.429825408984810	1.453069846298220	1.431850937738420	1.445385058568040 1	1.426058583453820	1.447766770639430	1.425496911465560	1.453565417874820
SCC	1.318963287718669 9	1.265506990455469 9	1.254146030404	1.258213628060730 1	1.223863435711309 8	1.316914577710480	1.310778191602730	1.269472644691520	1.253861413533820
TASCO	2.507228804399720	2.541667318250410	2.941809784004820	2.695891379412730	2.626849605148290	2.770243925785530	2.506218169985730	2.534708599347250 2	2.951373004771780
TCAP	1.217199306857290	1.205929951948060	1.255005836246420	1.191866082475090	1.205442190465440	1.178031905639220 1	1.215799800022340	1.202471137146790	1.255013976176690
TMB	1.837667065293820	1.773172850638169 0	1.880512562594120 1	1.859131264719530	1.752721965007239	1.948330246565240	1.835732095912590	1.772454683391330	1.880086869169469 8
TOP	1.653679444733130	1.549340312873590	1.744702952342420	1.560922354515720	1.509707338321 9	1.573566275173600 1	1.632305862242180	1.548569764899489 8	1.744410829405370 2
TPIPL	2.313760490705920	2.066155335897250	2.791222032319690 7	2.526038515549280	2.400899871364560	2.525508064742729 7	2.302202154000240	2.059488467388010	2.790534992357000 3
TRUE	2.235488600153120	2.239734157329990	2.604556329154690	2.398091804278059 7	2.324246677652320	2.322022668333790	2.236382930864850	2.245750687875010	2.615928959672720
TTW	0.945160899269023	0.926159055622330	1.117986094479690	1.052032467726040	0.973096877956670 9	0.951860296189110	0.946108703387370	0.924946782356050	1.117797687073950
TU	1.581730466745090	1.537293955316930	2.219493070139449 7	1.698500861591180	1.564414573818749 8	1.729702421328640	1.580962563208360 1	1.534567095278470	2.219959163614470
WHA	1.875795551172360	2.084778908988760	2.561304547556550	1.950541178207010	2.049908129221090	1.891385703994790	1.871258513158479 8	2.081982255778890	2.561506809879640

PredictPrice16 : MACD

Table 46 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	31.006605828136102	31.557126893937998	26.2446049478329	30.309407658539598	31.8527308049801	26.388638127858 5	30.898412526856 7	31.653435671449497	26.99453543694928
AOT	36.581333699587 9	34.624424152872 8	19.842701762150 7	36.878143978868 7	36.290537067916404	19.990840642015 4	36.149182011506696	35.20676058795 34	19.40938913333838
BA	1.850550644027 94	1.917739236482 14	1.116899227895 01	0.0	1.830650863262	1.683561557235 06	1.801671344972 69	1.88546834252 54	1.114743500800 51
BANPU	36.692747245661 6	36.885062631159 8	16.549891973936	37.239723250913 5	39.492749677051 4	18.511118477319	36.264129642265	39.623600115558 3	17.48099260635 79
BBL	6.5257228750728205	6.934921943783 47	2.972939987340 69	7.069141886926 44	8.131320171090 82	3.001271226609 06	6.945302503463 12	7.590432288128 37	3.023134787907 53
BCP	3.637338935980397	4.234137910935670 5	1.952371103040179 8	3.632844303053068	4.179189833735 03	2.088008834170 61	3.578361495972 57	4.209508572408 72	1.926744730290 501
BDMS	20.579278072697697	17.940412022028 6	9.715096686071 64	20.212794395183 1	17.821580639710 6	10.656134754379 2	18.089815118945 2	19.278053131478202	9.332025961168 14
BEC	34.197507698594 4	37.208373637931 6	13.093990203055 201	37.211718201564 096	43.509172922282 6	13.161388925702 9	34.691441795207 1	35.808547449989 4	13.313674040593 6
BH	8.7705224908313	19.184464579097 7	2.312335065219 91	13.633302929215 5	16.628669502479802	2.300715566756 66	11.819132330392	23.352758906108	2.324397209798 34
BLA	23.495181029024 8	24.113982650046 7	7.973167569359 6005	23.937131226209598	25.429246420739	7.812790354036 91	24.148728464433 9	24.327867347942 6	7.872291223706 33
BTS	1.363125211891 82	1.598086159259 799	0.950390831976 72	1.374815057203 89	1.732692606675 27	0.972906657610 482	1.454813514061 4	1.360473833009 044	0.954360968475 5939
CBG	31.677113308332 5	31.643230703833 2	28.536091538482	32.302888048712 1	32.167238983492005	30.127974254742 6	32.160716157259 2	31.699854598990 1	29.253438744439 503
CENTEL	15.159033556024 17	15.752177583031 2	6.763819384094 37	15.886834702642899	15.923865934334501	8.105543221876 46	14.120291020890999	15.838179419125002	6.698861586236 06
CK	5.342001521139 28	5.871388399440 36	5.148805359053 47	5.983355354803 92	6.515670886086	5.832652038633 73	5.331384069201 88	6.503902950278 85	4.737872066693 96
CPALL	3.540819075808 22	4.872677155096 61	2.076802757428 05	3.787682554132 3	5.534835230322 6	2.427001213142 03	3.995631629214 68	4.872629486643 9	2.076128038177
CPF	15.569344821014 8	17.150631386464	6.199805595700 799	16.922154200311 3	17.596321214290 7	6.316514872897 661	14.339708358338802	19.133626351435602	6.232080502682 21
CPN	12.696862338791501	12.470296586361 6	10.846593207012	12.664029688969 7	12.553729096919 16	11.632758046643401	12.537069983084 4	12.426731046642	10.597925135158 6
DELTA	3.183775416147 99	3.375691079305 87	1.806383144588 76	3.131616779409809 8	3.050688977752 83	1.803034843341 56	1.869402183126 56	3.712432988607 18	1.804879320372 699
DTAC	127.943518470817	133.00200731630 5	59.413959570899 7	128.596636400172	130.133424086806	62.890492709829 4	129.618536680643	134.893271606816	59.918749788757
EGCO	10.232251372310 9	9.522790430703 72	8.549777699320 39	10.430668242751299	9.7856695467058	8.438134502881 54	10.277815459865499	9.791654527535421	8.289125782697 12
GLOW	1.608297047099059 9	3.325832776076 58	1.425766551362 23	2.232610534911829 7	3.520264373320110 4	1.504146092972 94	1.783052783327 88	3.030246447573 94	1.423259015279 08
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	8.391789577649709	9.043973362027751	4.212143230885 91	8.416936492910 39	10.933411296511201	4.408564171078 11	8.098216536778 89	8.589274878825071	4.101065475002 28
INTUCH	24.58745717230 5	25.926802068573 6	22.039041945608 7	26.870925700266902	27.161724861131496	22.334369049809 4	25.561541826875 5	27.305030972907 29	22.689148224499 8
IRPC	18.251728398158	18.448976019525 603	9.303165838234 96	18.337513382495	18.809439585676 1	9.217396889084 58	17.734778240970 8	18.581428566263 2	9.387853318433 62
IVL	10.698330285408101	11.295086910042 6	6.191429100977 3805	10.888416251538 2	11.255511358246 1	6.350200263050 31	10.690770764398	11.267825391136 3	6.211425828698 75
KBANK	2.704442263945 48	6.255249540671 021	2.068218770737 23	2.902146979867 02	5.899303036359305	2.406411952673 36	2.255336492289 02	5.861995447404 13	2.102039572236 698
KCE	37.915052220293 1	39.621037166087 1	20.252531004869 002	37.264269975440 4	38.585118959304 2	21.135242820579 7	38.545475006756	38.047319088363	20.182901413946 4
KTB	1.968999038570639 7	7.255179505309551	1.186483086856639 08	1.350732844084 46	3.297484525339 91	1.235380430309 62	2.353042528653 02	6.605040687911 97	1.186240843696 96
LH	1.449782110584 11	3.085674325019 85	1.249474930488 2802	1.556854302675 33	2.843529367430 65	1.225758396138 73	1.459848958345 59	3.016038023899 09	1.247085633782 37
MINT	28.579830416362 7	26.674933206354 8	15.236777454603801	28.488618491363 6	26.242737878989	15.600968068984	28.706381954767 8	25.143520949585 09	14.280637663098 3
MTLS	3.914773274889 68	3.968217512911279 8	3.567397399260 85	4.059709803715 16	4.052619075321 89	4.015662417300 93	3.452687304197 965	3.971746851141 697	3.565648833407 904
PS	1.474339065645 924	1.927146982858 84	1.392713229658 1801	1.720919902218 102	1.736533402138 07	1.460450362324 66	1.842661509183 398	1.763357694967 59	1.390367044555 75
PTT	8.069170226220 77	9.470166131683 11	4.328939256258 5	8.352769402955 321	8.130796889564 37	5.020678620860 13	7.659065708285 64	10.19724694821 9	4.402830816510 87
PTTEP	35.336416351815195	60.881932168596	12.029010616160 4	35.463446340483 5	51.123912096362 8	11.861154052200 1	35.555862978571 7	41.689848504885	12.050137112764 6
PTTGC	9.910708162476 57	10.353093277917 1	4.249103752989 62	9.706478841353 38	11.899846568206 2	5.743148626976 05	9.820768458107 15	10.333114211289999	4.262546015952131
ROBINS	7.708002039003 6	11.559003633813 3	3.788677579238 6	7.746902252443311	9.584974344857 44	4.216783143061 38	7.883255064010081	9.260643439073 69	3.818644351760670 3
SAWAD	9.011652142174 88	9.442742876265 69	1.924207280168	10.051132031108	10.172067498986	3.132442487911 01	4.897729729043 13	11.025605369096901	2.030923967310980 2
SCB	9.344545169353 2	12.159090017223 1	3.917739792076 13	9.703068380572 71	11.790079814743 2	4.111382547419 69	9.154402499743 46	15.437289502151 3	3.957078121069380 4
SCC	1.192607103823 39	1.427356347671 72	1.210402249044 61	1.282698497020 58	1.952885354160 05	1.181680185350 54	1.346149325971 32	1.931234774959669 9	1.213034774784 89
TASCO	81.32342819211 34	81.32467198428091	81.684773528618 5	81.284960540567 6	81.362041668229 1	81.639101936673 1	81.344727025112 9	81.334604032002	81.655183227829
TCAP	8.852401573494 86	8.636805125511 5	5.121055362140 69	8.858173730092 89	8.710670117555 58	4.937035309572 909	8.737250327468 74	8.586989240145 54	5.046931035944 549
TMB	2.418164327120899 7	3.529503658741 14	1.538034129974 77	2.692411431520249 6	4.404282421319 38	1.450594745467 53	2.420575498765 42	3.507492467155 5	1.507186343904 83
TOP	23.646183757521 16	22.496055408955 3	20.593218905825 5	23.780241184128 3	22.722048218394 2	20.509652083771 8	23.356910750136 2	22.054044717471598	20.025436492295
TPIPL	2.903694820235 56	5.419166936008 99	2.545220926346 46	3.334990805720 24	5.531018552686 46	1.961408888324630 1	3.033143448069	4.027284953207 41	2.356410265389298
TRUE	26.737030671191498	29.223204718435404	2.445158043803 27	25.638024374971003	29.227816989891 3	2.218197822764	26.703591890934 2	29.168447801490 1	2.458053286382 05
TTW	1.537830958949 79	2.941531343364 36	0.989346009178 79	2.916750428655 62	2.760493691944 17	0.939074749510591	3.426208805113 34	2.723314650291 98	0.989059550716934
TU	1.651092536624760 2	3.070365894621 27	1.331695314502 49	1.558708702174	7.349659303954 72	1.304741430406520 1	1.540397778001 34	2.230203550089398	1.331155334476991
WHA	1.575803242852109 8	1.942366857282 75	1.780401672670 32	2.632108555408 28	4.689778586891 94	1.862106097122 65	1.767021699279 44	2.018114284068 46	1.780394203356059 8

Table 47 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.81897646414369	1.81393380566604	1.7449369556461598	1.72378966415523	1.7247821255203002	1.763809385459006	1.8188062274774	1.8241006219411	1.7440819632855702
AOT	1.0380268076199701	1.05392141459121	1.037741148666958	1.06306284566737	1.07707498352119	1.0652163175863	1.055810802992058	1.055160546495054	1.03773907008584
BA	1.041588410069933	1.03746231105365	1.040652755619332	0.0	0.9821404289693161	0.9341569644456279	1.042348700331021	1.061217063107254	1.04244472040619
BANPU	2.044940488892957	2.11111547496857	2.011558482710979	2.148585576888685	2.14709619514185	2.07478019521296	2.007159854558903	2.085639707676634	2.0115322576448698
BBL	1.2197227787398002	1.2282425779852901	1.223793581101159	1.25541390765088	1.2470913994348198	1.2353655439714202	1.217357144410 3	1.2282706565411	1.22232783516157
BCP	1.386184742946811	1.4162342226390199	1.359235512535776	1.4135132470179799	1.40989024151385	1.405399241833 06	1.386268451048449	1.416343863031 42	1.3595983502008901
BDMS	0.951665239635803	0.9832269742010661	0.880680499854288	0.958026200952772	0.9830453410178 66	0.9054313528478 06	0.949274256676 41	0.9768220510938881	0.880699045061106
BEC	1.810813158240 49	1.81199521691644	1.687854403614 63	1.739493135532 84	1.7081738994068	1.666829990818 79	1.812566471235 21	1.811198704202 1198	1.68723456974666
BH	1.4397527077543	1.44885210859096	1.393192954633 25	1.4546329042422	1.4822112765575501	1.4618052760143698	1.438838952886 32	1.448898696454 76	1.39317650438794
BLA	1.645890606025 84	1.62733783444057	1.592593892980 84	1.69730484244343	1.69246940151847	1.699612319750 07	1.646629239692 34	1.627620795037 5599	1.59277770969536
BTS	0.9599931651049051	0.9691690276212 66	0.929563757737812	1.0316513190937	1.0179155842443202	0.9988647572988 32	0.9600004365853 05	0.969111282251335	0.9289094255602 79
CBG	1.983807342901 22	2.01613692331861	2.001720161878 93	1.96428006602989	1.9665440197295 8	1.954430743523610 1	1.988470650160729 9	1.991661362164260 3	1.9926987695152302
CENTEL	1.644565833653 52	1.63894336417059	1.551451457886 82	1.6383927117870098	1.6272473178051299	1.567498039042580 1	1.644400479083 34	1.639146152370 12	1.551924240686 963
CK	1.386158556819 04	1.38473179826 8899	1.369456612059 48	1.5062631277435399	1.5098336605149 7	1.338312812092259 8	1.385594025996 95	1.409931951247 17	1.3696617577901
CPALL	1.350905996343 195	1.36794231666234 99	1.326300048675954	1.4094965988176 8	1.4140529147896401	1.355882871714 7	1.350923532413 71	1.367786606874 85	1.32544494828328
CPF	1.997197219171 5401	1.98421735611509 02	1.979492865225 61	2.1305829876032	2.0371383463124	2.069565265494 9397	1.995806981722 09	1.989285862739 62	1.98016009938165
CPN	1.267033157626 76	1.26066467913 918	1.225173163578 39	1.3240522722432 9	1.2987224273747 2	1.262766092073 2702	1.267053435115 59	1.260108646850 19	1.2247736583528301
DELTA	1.814959633232 5098	1.79539944766459	1.716839162630 5801	1.843948901089 67	1.85311633075836	1.719834154060 55	1.798257588422	1.795457045629 5801	1.71727962681091
DTAC	3.301625087749 25	3.26924650883197	3.105720544367 38	3.13715022026762	3.1467871275659 4	3.096069141811 322	3.166206190777 73	3.268768202072 21	3.102178615102
EGCO	0.9316679182058869	0.9367982857590 86	0.894766946926365	0.9603077702744749	0.9769617433707211	0.9987368404603 96	0.9314078499080 17	0.9655799732938 65	0.894780637131108
GLOW	1.393692429004 98	1.42113066066396	1.280594465097 02	1.42772067442795	1.4006277041305901	1.349346471016 061	1.393543943116 19	1.421307364419 94	1.2803464636383
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	1.493303064634 25	1.54421166535554	1.461783295426389 9	1.495356102377 96	1.4866485520980501	1.464340511556 44	1.537223520349 9	1.559067094073	1.462036800205 43
INTUCH	1.639729355583 67	1.62407856720769 98	1.606647892834520 3	1.6211316467983101	1.6300680985476501	1.620868064787 72	1.639529182160 6601	1.624118166654920 3	1.606058909458 85
IRPC	1.32891292213 84	1.35001832501899	1.293750929957 9	1.3038009178968901	1.3092556076042	1.298007925485 11	1.329035668380 63	1.349537865592 34	1.29369201827949
IVL	2.231724019699 78	2.30664133592989	2.134403338876886	2.0693335714831 1	2.2420442577286 7	2.173178122137 47	2.188587360285 85	2.313988623392 75	2.133906313017 75
KBANK	1.4306078529653699	1.50053188948889	1.393849441532 22	1.4706765253717	1.4817265794332 1	1.497841940964170 1	1.430594777599 4	1.500560508205 52	1.393828455068 08
KCE	1.454915856733 56	1.47604158177689	1.444563290719 56	1.4773082149722 6	1.4882806955295 7	1.453429388884 681	1.455213437061 3499	1.465256178664 74	1.44445084745866
KTB	1.180870097101 09	1.18329992978838	1.151074933425 18	1.1890028007143 2	1.1925625366279 5	1.180456794639 5899	1.180954913031 44	1.183072859310 2	1.15122124169238
LH	1.325420991623296	1.36871487586718	1.254197853568002	1.3405151646571 4	1.3690300764350098	1.249106554348 1699	1.325774659555 48	1.368149904026 593	1.25442082366503
MINT	1.688862452728 11	1.72185590914627	1.678373491603 66	1.6294248674204 8	1.6831226833170 498	1.608657523 06	1.705480121862 3	1.717725157559 75	1.6711566829436901
MTLS	0.9780243968686861	0.9580789056260419	0.950521586656324	0.9744754394678 25	1.01421021120829	1.018513990177630 1	0.9784882101617 32	0.959441807581644	0.9492205019861489
PS	1.342329055952 35	1.31757584921645	1.233502959388	1.4695658212994	1.353811276901 55	1.204784902146 72	1.287402260147 99	1.313912108133 94	1.23519955911927
PTT	2.009875481264 48	2.04241451736626	1.841979571396 55	1.872834430726	1.90446776304738	1.802400221685 45	2.009874489753 74	2.092874107916 34	1.84146108608008
PTTEP	2.356190600551 88	2.73691616373335 97	2.235654319330 26	2.297302300894 18	2.319959984515 59	2.200143718060 99	2.350634201000 32	2.841138693206 74	2.24820095840083
PTTGC	1.888914826771019 9	1.91132396087771	1.824053111718 17	1.889149078442 93	1.8508950695737498	1.899854158772 38	1.896958554405 4	1.899891804109 92	1.8242326657025
ROBINS	1.559205727640610 2	1.58918382928723 98	1.442846730100190 2	1.463252611356 16	1.4891987654093	1.455048797610 67	1.559210585457 99	1.589168998542599	1.44280572160401
SAWAD	1.565109224750 61	1.59050618682975 99	1.421108096594 57	1.464077435266 24	1.4147726928332 5	1.383167356479 46	1.552561798405 3101	1.588337688904 8302	1.41632714878244
SCB	1.518553394504 53	1.51767582515486	1.425089163820 71	1.526364157328 56	1.5115045250581 6	1.444399778942 36	1.462749383493 29	1.533906599406 51	1.42293710682975
SCC	1.100079406380 06	1.11538057110312	1.089699485914 85	1.116368391819 28	1.1473788096318 1	1.069106919235 91	1.100269402781 11	1.115544076171 33	1.08739488257123
TASCO	2.306261978921 58	2.30342096244798	2.290726325278 53	2.309874371796 7102	2.3050688824085603	2.328656239582689 7	2.305506779596 9	2.301682775444 52	2.29156532410381
TCAP	1.284821221655 77	1.27991652549348 01	1.177909699571 53	1.1710099100286	1.1758995529293201	1.129600431177 97	1.281747866621 39	1.276521690695 67	1.16246095481338
TMB	1.463102670311 13	1.44763277899288	1.364902879549089 8	1.474103289613 53	1.4943243749583	1.513247518389 41	1.457753394185 7101	1.447659575001 67	1.36485591672463
TOP	1.474651755126 9	1.50755063894259 01	1.521175302959 79	1.541677750279 42	1.5548195502943098	1.510156273707 96	1.474873562301 92	1.507348225016 0501	1.52098636724061
TPIPL	1.734872507150679 9	1.74823353974092	1.642618663853097	1.924334170023 29	1.9472335704100 7	1.751303455892290 2	1.733553090225 46	1.787214725997 38	1.63625326833937
TRUE	2.104540765281 04	2.09448241049509	1.977088715409 86	2.269119032071410 3	2.191481907216 72	2.059754049943 71	2.032102875807 02	2.065127969429 06	1.9753963114648598
TTW	0.8469763357491811	0.8503086248338 65	0.821603058937978 1	0.873981641097 11	0.8487209372479 73	0.8012306102074 96	0.8873894038881 7	0.8537892648737 79	0.8213156458362 64
TU	1.405122703749199	1.50940016216479	1.312191378209 64	1.398684461310 52	1.4691202525107 9	1.339049930932 65	1.357124876430899	1.472348273085	1.30898671768406
WHA	1.819317283483 89	1.96884088377142 02	1.595579845099 87	1.725498030135080 2	1.853184237969749 8	1.554300835039 11	1.817927181280 08	1.968821231655 69	1.59551810548799 99

Table 48 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.800063597193 51	1.799881826704 56	1.791955687127 28	1.749247479700 3998	1.745792163271 73	1.786213977140 42	1.798142785261 6398	1.799066227818 37	1.784423123501 8202
AOT	1.118323971267 91	1.195039824590 998	1.083439560656 267	1.177903060953 29	1.242822069977 99	1.058252821671 09	1.040163902527 53	1.148700264122 42	1.03827796731824
BA	1.041627362436 29	1.060999702215 27	1.040652755619 33301	0.0	0.9688120739035 13	0.934156964445 631	1.012034979698 5	1.019059095608 71	1.0424447204062
BANPU	2.022453519725 66	2.021619196032 116	2.011584827109 79	2.065428876077 19	2.129948925717 13	2.074780195212 9703	2.022957713321 71	2.111529388567 28	2.01153225888 211
BBL	1.199435438432 98	1.226290251746 17	1.229036321448 33	1.251323850329 17	1.242330014491 11	1.237708988677 14	1.199453604176 9301	1.226238966519 93	1.22742143163391
BCP	1.386184742946 81	1.416445157001 27	1.359235512535 77	1.411861104776 0399	1.409941434518 75	1.405399241813 06	1.386344556844 03	1.416363788207 8801	1.3595983502008901
BDMS	0.951982891702023	0.976719732675305	0.880692723594826	0.958125572785542	0.982810998519649	0.905431352847804	0.949279694265341	0.976701907956 4809	0.8806990450603949
BEC	1.759050746586 44	1.765099507569 858	1.680039722645 09	1.714451441211 15	1.693284690602 5998	1.668004616241 03	1.757976223140 2798	1.763903032517 53	1.67979411645440 01
BH	1.428933041889 81	1.470365995710 94	1.402883024643 33	1.467012459181 61	1.484439340619 38	1.470069858063 4899	1.432546577639 2101	1.470288762099 49	1.40274053569314 02
BLA	1.646474486211 258	1.627216519702 65	1.592596925080 16	1.695454276279 34	1.691245965857 28	1.699612319750 07	1.649112243535 31	1.627111067528 0798	1.59277709639649 8
BTS	0.954038915136203	0.964763556771749	0.929606716299241	1.030557952893 8	1.017372931647 75	0.996322152173329	0.953479768719 4171	0.964492110563137 9	0.9285550751383469
CBG	1.968859755949499	1.990196013524 81	2.001720216187 8798	1.964572657400 61	1.969862688044 99	1.954430743523 6201	1.983263941885 998	2.014432806776 4	1.99269876955123 02
CENTEL	1.666669920028 27	1.663390564404 0798	1.551733467317 334	1.645527711072 17	1.636556582791 44	1.567983280601 23	1.676915344560 1699	1.663780713070 59	1.549748353846 529
CK	1.464882968413 03	1.511101351105 16	1.385256463853 85	1.455165552881 68	1.475243314224 36	1.336992732374 05	1.463220761892 76	1.464250523848 28	1.39795179510814
CPALL	1.350905531763 39	1.367760906835 39	1.326300048675 953	1.409473323000 32	1.411004984603 01	1.355882871714 7	1.350923532413 71	1.367722632057 3	1.32544494828328
CPF	1.966887082905 09	1.963788050047 2702	1.983090362148 1801	2.017893706906 94	2.139051148259 49	2.072218178787 92	1.966295262941 28	1.964483729310 32	1.98341735522176 02
CPN	1.267169708442 9	1.260713254905 3501	1.225173163578 39	1.324182418742 56	1.298732112746 2002	1.262766092073 26	1.267061939964 6	1.260108646851 09	1.22477365853283 01
DELTA	1.814713270189 4502	1.785961428448 49	1.716839162630 5801	1.843877875552 8082	1.875124819777 869	1.719834154060 55	1.798403348959 1802	1.795399264083 98	1.71727962681091
DTAC	3.126774263463 27	3.102678354140 81	3.097555046454 1297	3.159042906217 6804	3.153397805171 93	3.081126895626 83	3.126999570361 3302	3.102053194017 47	3.09676153797062
EGCO	0.932385229079957	0.964569989528614	0.894575336187 9101	0.963727785847 811	0.977090335365 954	0.997710999031 956	0.932281065472 617	0.962972777596 535	0.8945814616959589
GLOW	1.393649033906 59	1.438160936841 38	1.280594465097 02	1.434701309792 67	1.400637517045 88	1.349346471016 061	1.323185544671 17	1.438092882098 44	1.28034646363883
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	1.495378921656 48	1.553329708973 33	1.459212314705 83	1.500699146598 3701	1.498103546400 67	1.459129146078 56	1.513736321317 1	1.553458179572 19	1.45941180170106
INTUCH	1.640051285063 47	1.624078567207 6998	1.606647892834 5203	1.655722023187 32	1.630025301188 39	1.620868064787 72	1.639438286367 6601	1.624068212596 75	1.60605890945885
IRPC	1.408132810920 9301	1.447524642885 2002	1.317610368496 5599	1.365485055661 19	1.414023808003 8	1.298031781495 44	1.407775395125 39	1.447272969741 97	1.31757655028529 9
IVL	2.190528036483 16	2.221791043346 09	2.198100904462 36	2.375023141934 03	2.309697477716 5302	2.197209373183 8398	2.190296928617 46	2.205053135005 83	2.19139763994775
KBANK	1.444077337987 75	1.443509932036 03	1.413377928198 1	1.491009320977 93	1.516847540193 83	1.505036585441 72	1.443788693555 38	1.459310317993 06	1.4086984146351
KCE	1.479653075230009 9	1.501268198303 25	1.454451392510 9102	1.530594742965 29	1.534965149245 43	1.456069240723 02	1.449918344980 35	1.476046434585 95	1.45390676602722
KTB	1.180889111263989 9	1.183276134667 72	1.151072168616 14	1.188948434837 78	1.192530170249 33	1.180456794639 5899	1.180890658865 25	1.183092583723 1	1.15122124530782
LH	1.318098710161499 8	1.341796284812 99	1.239895358745 27	1.297658611398 3	1.280921783653 1	1.252698482299 17	1.318425246167 5001	1.350333313572 374	1.23998850267269
MINT	1.677042971088 22	1.690861287987 0202	1.688707337097 38	1.675729107554 29	1.720596689038 8402	1.670308666625 601	1.672203823811 01	1.685078502535 602	1.68199329216929 58
MTLS	0.957339150027549	0.958025126508961	0.950521586656 3259	0.953500888715 793	1.014440953306 7	1.018513990177 6301	0.979012484801 454	0.959792125532 396	0.9492205019861 48
PS	1.310732169801 52	1.411002495267 502	1.262108565151 33	1.379660707585 924	1.327836376193 3	1.233795196864 8	1.402317736460 69	1.409435863519 84	1.26227798498878
PTT	2.013790759023 56	2.023063019200 91	1.843476656810 6602	1.874387120417 84	1.900804151472 1599	1.802825097127 55	2.013540587446 22	2.023034881827 51	1.84274584780154
PTTEP	2.248793194929 37	2.328880181591 3	2.201413624677 93	2.180396450190 19	2.175147457002 597	2.249847977832 19	2.247700377726 1297	2.313459720126 44	2.20159831232395 97
PTTGC	1.888914826771 03	1.911268069141 26	1.824053111718 17	1.879865577984 22	1.850874761348 47	1.899854158772 38	1.888606981833 496	1.895435631057 98	1.8242326657025
ROBINS	1.55936111457	1.589129547415 86	1.442846730100 1902	1.463352809710 5801	1.489174774497 62	1.455048797610 67	1.559474229797 4	1.589127768840 4801	1.44280572160401
SAWAD	1.577500886050 23	1.590192787305 87	1.421108096594 57	1.463962324620 05	1.419782029587 8298	1.383167356479 46	1.551421226088 46	1.419901481405 9001	1.41632714878245
SCB	1.518471900370 29	1.533533524961 78	1.425089163820 71	1.502854907009 36	1.499483083736 42	1.444399778942 36	1.462537927171 97	1.533899390344 41	1.42293710682975
SCC	1.092614993583 57	1.108092358325 402	1.090231729401 24	1.130841632179 3902	1.154305512665 47	1.086567402502 23	1.092514029990 01	1.107804859740 29	1.08896106552184
TASCO	2.542586751062 86	2.501877215597 98	2.385949154047 87	2.456208906704 35	2.466681990518 45	2.490874099469 78	2.556682805309 01	2.508593067309 72	2.39811906453658
TCAP	1.284821221655 76	1.279857583354 93	1.177909699571 53	1.169843064247 94	1.175899552929 3201	1.129600431177 97	1.280266443332 9	1.276521690695 67	1.16246095481338
TMB	1.460256809169 1601	1.449524767607 46	1.370462656000 626	1.473823084067 64	1.504818159604 5	1.512110914427 46	1.458262475247 3201	1.444043233529 19	1.36978900863283
TOP	1.474661252893 61	1.507296308848 68	1.521175302959	1.541667465168 12	1.554769716926 98	1.510156273707 96	1.474872043909 37	1.507286576135 601	1.52098636724061
TPIPL	1.708662020043 83	1.795182765054 73	1.608929766845 34	2.201775137701 7802	2.130876560313 5797	1.879313594623 04	1.727619130750 56	1.775339623103 7	1.60649614977260 98
TRUE	2.070914748958 49	2.026316113936 42	1.987394715580 2902	2.130691543703 6897	2.079519950217	1.979729918880 8002	2.070097795909 2	2.036716901625 6	1.98690934053421 02
TTW	0.8349591784380739	0.832745869633899	0.818367671665 7681	0.879674178822 6641	0.891905487278 482	0.783176469913 8689	0.833490115824 161	0.832269895556 274	0.81744270731 3807
TU	1.357054470853 24	1.374821667354 02	1.312191378209 64	1.403177722760 901	1.478211972131 84	1.339049930932 65	1.330850721360 97	1.437185593671 67	1.30898671768406
WHA	1.844987984652 3298	2.010607583368 12	1.605375756910 4402	1.681291703432 12	1.956107362792 49	1.624615057646 43	2.026547308258 01	2.016128096202 84	1.60515351102767

PredictPrice17 : EMA Change

Table 49 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	31.9719513977414	30.6096013832063 96	32.6216435427151	31.6850679422741	27.9801782448280802	32.725522070815	32.3424571778462	28.7566530524639	32.8976893329733
AOT	27.5222042348918803	22.37597231825 95	18.9372943318263 97	22.7873365538734	16.8674171384962	19.0111903209865	24.8368822281175	20.286027172371902	18.912605668404
BA	1.8150773535352798	1.12994620583393	0.9653743786942 41	1.3820985648296 1	1.4376090055267 3	1.4358696892894 5	1.7461365601182999	1.1377171586421 3	0.9648235761736 45
BANPU	41.628276693948	30.388923750941498	23.8583978185967	42.580649552756	31.2928605278438 03	25.804294325565703	42.8514294800411	29.4070581160915	23.9216629215743
BBL	7.624893271534 14	7.74181417675312	3.8837629542404 4	7.7259500384959 5	7.3400220539295695	4.69621546384949	8.808517586067247	7.225685654108591	3.850391692020 79
BCP	1.859185072465 87	1.926761841686 65	1.7650498318861199	1.882096414606 42	1.9254801122240501	1.8424013538847699	1.890951406091 98	1.898455058503 97	1.75927522525375
BDMS	10.0505851074826	9.2385566290726 2	6.3755332341074 5	14.3265860633736	8.7753635848672	6.1243827191312 6	10.3740140537134	7.967496835636 06	6.3043857497542 41
BEC	29.7113270154718	27.4853735348466	17.0318357091236 98	33.0424326544320 05	24.9961947728892	19.769816828689	34.7005445917287	24.1012348770224	17.0453529055009
BH	7.523752296996999	10.3327721327897 99	2.3669745864037 3	7.965584223770 67	10.375645298029	2.1951545909453 9	9.3259003305856 1	12.058631136457	2.1524913257895 9
BLA	19.878296323405	13.7450495715655	9.7540027335232 99	18.5120118830191	14.0689624368373 01	10.458654171096 7	17.756827159487102	14.596024148516202	9.6644171219417 03
BTS	3.553907075250 22	2.5926669563118803	2.6329405299761 1	3.8667352409946	2.3828418655208 1	2.5098265552482 1	3.537020177494 68	2.417150783415697	2.6512696374135
CBG	27.451240899040602	27.424157123864003	27.2138675336437	27.4296640445616	27.8933284687657	27.898849242805902	27.735869814252998	27.753800952441498	27.169776989898498
CENTEL	13.040125800296 6	8.871231386132 35	6.7868516796218 21	11.8445081244268	8.5250464161474 9	7.4617164253473 1	11.651431975830 4	7.697915509367 12	6.6258890747540 2
CK	4.022661569385 79	4.713653827213 59	2.5341009496707 4	4.4008679684007 7	4.0606840475636 1	2.779120492915 2	3.9353569253105096	4.047378248494 38	2.5146889855151 7
CPALL	2.7558207805990 8	2.4538949036549 3	2.0291325195436 4	2.8259215349831 4	2.2600232854172 2	2.1169938303489	2.907748680558 3903	2.270985218360049 6	2.028748712359
CPF	5.848964662953 95	6.615797514781869	4.0464175916459 3	6.3406513344867 8	6.6155470043911 39	4.343373043888 19	6.122969673413 71	6.777178580609 81	4.0852066618936 2
CPN	10.524277845665	10.134864926298 8	8.7778307668517	10.415843740366	9.4133108147 4106	9.4845166431874 9	10.431147999967	9.358706331092 47	8.6943066276716
DELTA	5.321553240191 77	5.184674149217090 5	4.1215899173107 395	5.5178940891633	5.2941526800478 8	4.6449259537518 2	5.883305533276 15	4.512905510897 93	3.9488914739301
DTAC	141.201518042494	117.47335913146199	86.8997348693093	142.305487904759 02	113.810938398475	88.2342433284185	144.717478191605	115.564902316524 01	86.8485987941509
EGCO	8.49901247065 76	6.45576721743764	4.7778423833066 51	7.3777347496706	6.6524706195871 79	4.9729481195037 5	7.654685082055 98	6.960471686594261	4.7996574379669 8
GLOW	2.748370884115 27	1.7507665413309998	1.7831489615963598	2.628561784588 78	1.9431889584128	1.9904513124304 6	3.021040366044 64	1.692815586263270 1	1.7851564261271098
GPSC	5.532502084444 21	5.021142721502 52	3.011463036224 57	6.5109652234904 41	4.7192975952493 9	2.6828449600472 2	5.922152196168 92	4.868999788179 07	3.2881585756992 3
HMPRO	5.636443217343 44	5.069591671728070 5	4.0324774113892 91	6.0472313346081	4.6665909678471 5	4.2302888419748 6	5.427483410566880 6	5.151279405846 46	4.0048931692188106
INTUCH	28.355768425036498	27.0874950413536	29.1761831248936	28.6294432531998	26.1026648828614 97	29.939962806499697	28.380137690941203	27.429004236112696	29.435038061145104
IRPC	11.271806041029 5	10.645310690408 6	6.5894580130256 3	11.360501350199 5	9.1644480883860 4	6.9860357430894 81	11.352821675235	10.87654659193	6.7600114326673495
IVL	9.6315390585484 65	8.435149419066 36	7.1566475772067 59	9.7142689809197 3	7.5318430435985695	7.271390671454 26	9.772553453693 13	8.112423157200902	7.0780477552167405
KBANK	3.864784433305 6	3.2799983697133896	2.9601036952510 2	5.0595637170103 59	3.4569312030902597	2.9418651793216 2	4.036576130020 74	3.487119746925 54	2.9508358435038 38
KCE	33.7368180760225	24.669053157369 7	30.8853973375905 97	33.6199179235405	24.3976199043883	30.975417389421 8	33.3300613424245	26.165492775196604	30.907859260806703
KTB	4.291015588962 76	3.351077466400 58	1.8753270815517702	3.7858059633513	2.9846070258832 6	1.9167560011266 56	3.7090073066389897	3.306781644135 59	1.881940326208 53
LH	1.629833760214 55	2.454744070468 97	1.3820654053393 2	1.8851496564 6573	2.7759396246115 1	1.6235766121 526	1.773865002834 02	2.506092039043 83	1.3809828597881202
MINT	15.991519018499 3	12.204680941463 9	12.038577407833 2	16.937324873277 1	12.271264648279 1	11.532577148676799	15.991470269518299	11.515816591897 2	11.862333368655
MTLS	3.574137988319 47	2.963595763473 28	2.5202438422915 496	3.9038599086638 5	3.252032201256 24	2.756387558816 56	3.608518036409 76	3.203487146868 7	2.5345198513773 4
PS	1.475228440921 25	1.578032400477002	1.460560717271 95	1.4915408644627 1	1.595502933755 21	1.470099296476 95	1.715561616575139 7	1.593668750875050 2	1.450644037717 6
PTT	8.992133119037 14	7.242030816652 45	4.613536583041 54	8.928409847914 64	7.917024560145 8	5.781277701937 48	7.877956124660 1	7.199771687152399	4.662070617343 13
PTTEP	50.301726896138796	50.087143562635 7	18.490015352323 2	47.919667080032 1	41.585836794321 4	18.562649306648 7	54.616324653937 5	41.559141221152 9	18.899844317651 8
PTTGC	4.730542467922 54	4.184247225627 69	2.828959717481 7	5.958097775897 08	4.589594648993859 5	3.108350489016 9	4.374649929379 49	4.480216089441 58	2.847754041771 54
ROBINS	8.757313443321541	7.766845173566 63	6.338612443029 19	9.279775365870 47	9.153036651011 94	6.779795782851079	8.945857732935881	8.046231342471 22	6.338536277711 61
SAWAD	4.78267980744	2.929882041225 65	3.378516209436 8	4.898344960463 82	2.988958099845 1604	3.324048852740 4204	5.310658860390 04	3.253822951283 3	3.341165730522 64
SCB	10.635900479841 2	8.548531430206479	6.378865507479469	13.631251130175 7	8.738832831947 56	6.779962809502 12	10.806987678926	8.915582175129 13	6.459795918193969 5
SCC	2.885900619786 95	1.736458640664 86	1.796172973996 002	2.274029767411 44	2.104976350666 9	1.895060154078 99	1.822032531698 4702	1.781028480299 1599	1.792419451272 15
TASCO	24.460165842818 3	38.202565229360 7	15.740961424099 101	27.822482712154 2	38.840507802475 4	16.461009834722 4	25.623190575628 9	34.19588164724	18.921309219574 4
TCAP	3.658620265484070 3	2.469065384123 1	2.173856726524 53	3.561851017013929 7	2.639821505965 9	2.498712404992 33	3.089053557825 12	2.813561125964 4	2.176992443587 3502
TMB	1.984402892220520 1	1.665518263302 64	1.498971546809 04	1.765364928552 92	1.732120642251 18	1.520177160768 72	1.848553963660029 9	1.642164654447 89	1.498477442076 56
TOP	18.815194591566 9	16.664165629 2	15.187361255581699	19.136329373820 2	17.045793869431 8	15.554057734552801	18.157664810494	14.334305739099	15.120750023464 99
TPIPL	3.823220402158 55	2.275740494976 41	2.443049195449 95	3.812549411382 16	2.596337257172	2.677103099433 31	4.614821474966 9405	2.284852456718 55	2.446128790734 17
TRUE	15.585053272178799	8.623480262517 77	8.792017862683 55	17.465656532254 4	8.702632236185 1	9.518953940240 2	14.918846369292 199	11.544519235046 28	8.763523243525 41
TTW	1.147148633546 03	0.849197672942 16	0.872855469652412 1	1.060053540239 44	0.849499650756858 1	0.880859232867 61	0.939793365840741 9	0.835688251357866	0.872803281249242 1
TU	2.827214905118 09	2.160990358794 53	1.563476727074 22	2.923710134404 21	1.974217559567 85	1.615507407121229 9	2.539409329659 89	2.373691828642 62	1.564888723869 87
WHA	2.760326265181790 4	1.772297611303350 2	1.968256891234769 8	2.004666903735 3	1.729459032498 45	1.970783749387 14	2.978084158280 1	1.816394828711 53	2.004985832079903

Table 50 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.902968235053 84	1.829706151430846	1.8396046132371902	1.8093014095263 99	1.800337767343 38	1.811348746874 19	1.905402311902 12	1.827278115011 92	1.833939322078 06
AOT	1.058341777335 32	1.040287902327 78	1.0362369215362899	1.0548012890459 9	1.0354969787224 702	1.022628491169 52	1.059343726665 7502	1.041127831403 1	1.036208676 9218
BA	0.989826022291009	1.049635250163 27	1.078183698430 67	1.043521094176 39	0.992246377145077 9	0.934583995088 7209	0.989003838594365 9	1.046283998623 42	1.078712981810230 1
BANPU	2.089792587788 58	2.082568063551 63	1.98843250462 43	2.169764845413 92	2.192941999815 79	2.190019513568 55	2.073589742750 08	2.094347341700 48	1.988331371500 596
BBL	1.203161732200 5301	1.189936976255 46	1.196682124150 37	1.229940861321 97	1.215687976342 31	1.216054420338 29	1.200902828906 41	1.190516155531 66	1.196864612841 44
BCP	1.444553370215 79	1.403523313640 34	1.360723766467 79	1.439598565395 12	1.416551058508 26	1.376853135435 94	1.454563011481 4099	1.398822796881 8099	1.360404247959 99
BDMS	0.908344063476244 1	0.889663290546 3335	0.877493692661421	0.914186933668 615	0.894396016661 448	0.910016650166 8011	0.893908703337 139	0.888037476847 158	0.877006733937 289
BEC	1.813561450018 6699	1.794475970889 07	1.694460109411 67	1.816882554832 4001	1.763137881342 07	1.760026009353 34	1.801667006423 24	1.814586439209 74	1.694241298249 7102
BH	1.435130650298 3698	1.404539514656 29	1.394270007097 11	1.422048920712 3099	1.446293994918 5598	1.428904152843 45	1.434522113559 06	1.404447616246 2	1.395284745363 23
BLA	1.742919422773 7501	1.683613801606 14	1.666851410505 6602	1.817392062763 19	1.790824510676 45	1.806807319211 1098	1.735174901204 9	1.681890994911 15	1.665650981443 08
BTS	0.990886544686 656	0.957753053738 321	1.00454029618 216	1.022017752578	1.014432573414 33	1.011986941462 36	0.981690977521 447	0.958006178211 359	1.003558965312 54
CBG	2.115626001725 0803	2.166491418084 7696	2.22955964300 63	2.080410371279 24	2.067368505771 12	2.088565843202 98	2.114665737504 65	2.159298557320 32	2.225850207501 43
CENTEL	1.558175738965 699	1.537156998059 62	1.517492400321 94	1.516732601530 3	1.503293972453 66	1.517352774284 77	1.550326923171 21	1.519535246466 43	1.517529750139 91
CK	1.375377142800 82	1.394447824312 18	1.360640771009 96	1.387207935394 42	1.387685775589 63	1.354783333791 05	1.375149593809 44	1.393568181449 59	1.360930843128 6
CPALL	1.384903475757 92	1.368760538504 73	1.338238550240 413	1.372545151655 57	1.359431575918 64	1.319419307099 94	1.384744563995 09	1.373133169027 47	1.337941220972 11
CPF	2.101932358416 79	2.074128803956 9197	2.052062689338 4	2.030827332299 74	2.012573917929 56	2.027913124643 73	2.103555662054 4402	2.077606763991 34	2.052247937652 79
CPN	1.284700846868 1	1.257791340582 35	1.253783803313 83	1.213018905090 94	1.214576614250 05	1.228705753725 37	1.273963916959 89	1.260007944032 56	1.253851683646 54
DELTA	1.829850792387 14	1.820844555392 6498	2.071366448033 43	1.808370994573 5702	1.836553081225 29	1.826800733514 14	1.808934072728 1	1.863549938 06	2.067966004833 8397
DTAC	3.172311361692 7403	3.158538826491 85	3.138459149688 26	3.308533954506 7103	3.223978458411 6	3.196565975212 91	3.171802245220 397	3.179013257274 8497	3.137385747899 46
EGCO	0.921171469804 949	0.909285249592 6909	0.898249769116 9079	0.927028387489 3059	0.907503052649 519	0.895238110638 5521	0.918248074192 717	0.908762462744 18	0.897847499280 355
GLOW	1.317949487473 74	1.264459273519 94	1.243830666221 9198	1.288250333502 33	1.296994654888 58	1.300302407504 499	1.324227465689 79	1.264237926394 47	1.243737876714 25
GPSC	1.37804842015	1.479141372124 38	1.568328180995 81	1.873447068433 9002	1.828420186501 3099	1.900958663564 17	1.438273928572 898	1.474403381697 5999	1.561517184592 52
HMPRO	1.550804660486 14	1.538867455849 99	1.525783458016 36	1.532052266959 38	1.509860632707 398	1.488069213528 85	1.558351214572 9899	1.536792975272 88	1.525063576809 68
INTUCH	1.662277775688 09	1.708314434264 9699	1.719007860118 602	1.727507481996 84	1.749158834290 498	1.757272525054 8	1.663270560012 2502	1.702667700399 96	1.714376906776 7
IRPC	1.358451292039 93	1.350964886193 13	1.337445958587 15	1.355708321834 57	1.353199339370 871	1.351354951041 33	1.365466013301 3599	1.344842903135 81	1.337543790434 49
IVL	2.125300663185 84	2.175179399668 49	2.135642550076 78	2.196637818832 45	2.157488867767 13	2.161198463975 99	2.144709833143 75	2.175330423944 45	2.135137380316 61
KBANK	1.463285751284	1.424518991517 2399	1.418138557105 91	1.384506455669 65	1.383001593052 51	1.401227683769 5299	1.456229413398 83	1.435392251750 902	1.417900637385 53
KCE	1.561629975956 3201	1.527909386507 16	1.530499783764 83	1.491740854189 12	1.472054417530 5101	1.486252978976	1.561462909082 353	1.523413356837 98	1.530122855570 845
KTB	1.200003758841 53	1.174228817248 14	1.161220691883 2099	1.246969663552 36	1.238150627545 51	1.194725840547 33	1.199924213290 15	1.179129518304 51	1.161103719279 34
LH	1.218521016051 7201	1.242566216921 27	1.210369450393 1	1.239401874103 54	1.225747650034 09	1.204960050369 23	1.218846924209 39	1.243615448086 94	1.210512211119 76
MINT	1.652855677501 77	1.628931775261 0801	1.567611240504 78	1.652016099655 21	1.616961118203 832	1.595859583241 798	1.634007615396 17	1.627315064161 75	1.567614346934 15
MTLS	0.944042038718 6029	0.959586320827 239	0.908918771018662	1.163052422860 0498	1.063903573304 11	1.136135387205 3999	0.948270740867 584	0.958943029725 858	0.909209077888443
PS	1.305667779294 699	1.302792534934 21	1.256363561926 77	1.263586820318 75	1.223939026351	1.223740600048 53	1.315570385762 74	1.308345370589 05	1.256591487031 9
PTT	1.904456952812 9902	1.872029237404 8	1.848712772139 58	1.851348821208 08	1.815649403188 2	1.824257044564 7501	1.929957261421 5998	1.878117025321 398	1.848112673719 8799
PTTEP	2.319650953326 32	2.300700128680 92	2.376305510695 36	2.289422723057 7098	2.274679019702 89	2.277627000078 6503	2.328081097305 35	2.300706017877 29	2.374757043012 2398
PTTGC	1.843275477402 06	1.821395416292 1102	1.811168964461 3	1.918723218727 699	1.875678444202 816	1.864933230132 7	1.856192385472	1.824266904573 68	1.810972106425 7698
ROBINS	1.456309266499 7	1.485288070233 35	1.488139561726 1701	1.472231853385 27	1.482578630628 2	1.481245863794 71	1.455458748227 2299	1.487946467062 9001	1.488008687902 8
SAWAD	1.547799700404 57	1.530004969181 5	1.520419832119 37	1.508764153504 55	1.510016859533 0201	1.504508750910 3499	1.579720349538 0001	1.517474041456 76	1.508537823469 0499
SCB	1.471573537889 74	1.442900665597 14	1.442650828721 296	1.546317834316 798	1.484231673462 4998	1.436270111860 27	1.472208557519 85	1.439408426005 0502	1.442407889790 74
SCC	1.094078820979 58	1.087767498754 95	1.128479306413 91	1.073609142732 06	1.076938912041 62	1.104786010594 95	1.090036563395 25	1.088177321406 6999	1.127952262015 9501
TASCO	2.433244468887 02	2.417941234184 5197	2.424840911438 96	2.474697400036 18	2.466117852468 88	2.459434861277 0602	2.435115864947 65	2.401304585984 8	2.416046663868 88
TCAP	1.168859149392 81	1.148428818556 85	1.151521931556 66	1.190603760301 4599	1.155599051268 23	1.139197105258 28	1.170059461771 94	1.149845824613 18	1.151856684445 23
TMB	1.514871845635 18	1.514517161483 96	1.420413023713 2602	1.533675090629 13	1.506937480214 22	1.523840614022 41	1.531217686921 2701	1.497487688706 3802	1.420713588640 8402
TOP	1.556096142891 53	1.552471285615 4799	1.541139763803 48	1.559713132982 35	1.557361728485 5399	1.569834247727 62	1.573487492691 6598	1.555182681671 44	1.524238764595 72
TPIPL	1.730861519874 899	1.712962090522 7998	1.628732280863 3702	1.866382164335 3202	1.846353117288 7597	1.859303730693 55	1.728064771360 8	1.715607055446 0698	1.628695485608 59
TRUE	2.109255695654 4	2.067743391358 9502	2.070366481015 6	2.132502116301 68	2.101608899070 0296	2.081229020435 2	2.115451390721 65	2.072482394209 5303	2.069915907038 4
TTW	0.844288675185 4841	0.837512402783 49	0.825267292156 238	0.856454217676 24	0.837014833253 634	0.824463222702 8771	0.828385791021 328	0.840401866474 662	0.825285207556 361
TU	1.417232861749 01	1.362516169382 1501	1.357508229602 84	1.427416361417 46	1.369333143545 87	1.332574486485 49	1.416853916591 199	1.362924979668 81	1.358216704750 4499
WHA	1.746240164643 88	1.686533686906 806	1.581787379127 8	1.863677007061 1999	1.876544321613 6002	1.818266739649 1998	1.748651422093 1197	1.679270625293 6398	1.579544319084 15

Table 51 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	1.79973381702316	1.77099731430839	1.7643358784953302	1.78841008935608	1.79257274184699	1.7777623792845298	1.7848143784252	1.765767121272265	1.76481323971278
AOT	1.0708885305787799	1.04514793329274	1.04270574675597	1.09321880087345	1.05857816614478	1.02505999900136	1.07021064022041	1.0457522686721	1.0427052793605098
BA	1.00216783049381	1.0523034007872	1.106970666623919	1.24625109794035	1.0346327911586	1.18051729580149	1.016181366688382	1.0561607892482099	1.094539009337538
BANPU	2.086312455550735	2.081128467155674	1.992783064586628	2.174985420888558	2.199403395659966	2.195960191959866	2.081478034133154	2.104914494044276	1.9920253374905799
BBL	1.1967922330712901	1.1908509289505 2	1.2016990321702499	1.22464764275688	1.215234306755588	1.21032502222664	1.19694014805086	1.1906667571529699	1.201734916123 49
BCP	1.448607303498 36	1.400351189816 51	1.360723766467 79	1.4448270281204	1.414003208804 14	1.376853135435 9501	1.446674415376 39	1.405284827175 27	1.360404247959 99
BDMS	0.902973419602 42	0.886589643644683	0.8774600935068609	0.914103971104 63	0.894350043960392	0.910016650166799	0.901158391000404	0.8889644532042871	0.876971555156863
BEC	1.774161899215 82	1.758537386161 38	1.679010036289221	1.754860598371 7598	1.743762585046 01	1.734122881140 3702	1.775800973148 25	1.740978083909 42	1.678855682440 0902
BH	1.467408698541 6601	1.416316082828 8102	1.409431405147 42	1.505523279074 38	1.543986672366 13	1.537937186376 78	1.464018782670 41	1.420291055355 37	1.409399000588 95
BLA	1.741207504785 45	1.682177983538 3899	1.666871638790 9701	1.751604179834 71	1.775141237719 35	1.806807319211 1098	1.739465521048 9303	1.692318960954 07	1.665776623969 1298
BTS	0.989372978181 5109	0.959715175469 627	1.014194600184 6198	1.026557843601 48	1.007676528813 7999	1.016536309107 78	0.989384065994 524	0.960438489082 2239	1.011698883192 44
CBG	2.114718581564 85	2.136911321911 019	2.22955964300 63	2.082609415311 13	2.053940796537 04	2.088565843202 98	2.112875293266 71	2.153693123690 5	2.225850207501 43
CENTEL	1.541890044728 53	1.539442156697 81	1.517908931372 52	1.514629591959 37	1.510353413418 6298	1.520590040127 7899	1.559575566304 95	1.526272306008 77	1.517721894662 44
CK	1.482006281507 23	1.482650541499 6	1.416483312791 06	1.389096930497 46	1.391345392323 02	1.366794675928 3	1.505217940633 34	1.487655555556 716	1.416248947182 0698
CPALL	1.384571203115 8899	1.373315083098 598	1.338238350024 013	1.380821525211 78	1.359742176625 04	1.319419307099 94	1.387908295650 9	1.366684558139 7902	1.337941220972 12
CPF	2.033395641656 97	2.016376121547 71	2.020190504200 02	1.988821195730 69	1.994403267941 05	1.987249185649 19	2.037336718434 78	2.014856088363 27	2.021070860121 1698
CPN	1.275448002350 7901	1.258048573969 13	1.253783803133 83	1.212842279903 55	1.213853238034 3001	1.228705753725 37	1.273681038605 57	1.257840480609 0999	1.253851683645 64
DELTA	1.820359059321 21	1.791263780775 4601	2.071366448033 43	1.832645497783 76	1.823615969701 8999	1.826800733514 1299	1.821228710476 4298	1.839436861326 6001	2.067966004833 8397
DTAC	3.145038648280 9103	3.136546101488 26	3.149660171733 76	3.197218200858 597	3.133411605712 14	3.156777779252 04	3.152591335586 12	3.138806334156 43	3.148021871017 0003
EGCO	0.920251230927 484	0.908851613469 556	0.897454285005 624	0.926551098757 78	0.906384935917 594	0.894047910767 513	0.918461626963 1841	0.907367420323 043	0.897020848983 9989
GLOW	1.339239392722 26	1.265590034996 5399	1.244657720258 399	1.269948270830 99	1.314460539134 43	1.300859581850 1298	1.316864679465 2599	1.268825200539 13	1.244560141704 1499
GPSC	1.376274498039 26	1.435308865550 05	1.568328180995 8	1.896185742880 54	1.821414731238 8702	1.900958663564 17	1.395618959706 61	1.452958631060 3	1.561517184592 52
HMPRO	1.568973626583 46	1.544754657785 03	1.536024776933 08	1.531137495508 87	1.509807330473	1.489942475727 09	1.558728495182 4198	1.544317121732 09	1.535196318370 1098
INTUCH	1.659513064015 43	1.710144236787 2501	1.719007860611 8602	1.717322832004 0999	1.732116988252 57	1.757272525054 8	1.660486193691 87	1.688526632775 8899	1.714376906776 7
IRPC	1.368371249027 73	1.343752041381 24	1.327250762532 94	1.380222460910 28	1.340251889911 44	1.370793389741 56	1.356266344941 72	1.339810400177 788	1.327315649419 11
IVL	2.181981989621 0997	2.178816060620 68	2.135055669349 38	2.156700390144 13	2.137903267515 86	2.146557602744 35	2.149668289579 91	2.175384866177 6503	2.135867126107 5
KBANK	1.479331090136 154	1.431344216914 62	1.437912466218 66	1.444393350450 4301	1.424730432690 926	1.436479219050 08	1.471624978729 49	1.427979871078 94	1.436795796431 85
KCE	1.573345071386 4698	1.535809138956 29	1.527249495731 03	1.496328997783 5799	1.473026514830 14	1.475806243318 19	1.573231124999 4702	1.538040512396 28	1.526866924962 5701
KTB	1.191678329909 5899	1.185423667027 01	1.161429764892 65	1.251407114455	1.240843669083 25	1.194736351811 371	1.195869209749 6	1.177246544707 96	1.161113125981 84
LH	1.210238717618 75	1.236062423765 66	1.211463669203 96	1.246296455819 77	1.237192340585 55	1.240266470405 94	1.211045313486 38	1.234554590935 43	1.211899441140 309
MINT	1.653920480046 6702	1.632479950549 895	1.551561959987 46	1.685507600690 49	1.680612338567 9799	1.615252748360 8299	1.656191725957 19	1.645064108813 1002	1.550352951 06
MTLS	0.952364858243 768	0.959228290015 466	0.908918771018 0431	1.142768231835 12	1.065060189174 9398	1.136376744468 37	0.944917897661 272	0.956005007232 12	0.909209077887 8211
PS	1.390688845331 31	1.399443466667 29	1.283854764685 75	1.275567325492 06	1.277066371359 0802	1.228091601324 4099	1.353048172784 07	1.420224666906 69	1.283954910048 76
PTT	1.901964288453 72	1.861244400148 8302	1.850717716475 95	1.845822190954 97	1.816680735148 04	1.824187428188 0899	1.901114491895 0298	1.885572155385 71	1.850166812689 89
PTTEP	2.249903124724 14	2.226286514546 76	2.270093050690 78	2.190962121447 21	2.202742214307 82	2.214454026872 6196	2.247228153761 02	2.228662774583 52	2.269911739417 81
PTTGC	1.844725498735 41	1.822140289668 08	1.811168964446 12	1.913836753044 5701	1.876072517013 2602	1.864933230132 7	1.856042071098 76	1.833872675675 96	1.810972106425 7602
ROBINS	1.456441714932 8701	1.487623388224 53	1.488139561726 1701	1.463841536673 89	1.482514102282 07	1.481245867394 71	1.456783427970 34	1.485757277932 62	1.488008687902 8
SAWAD	1.580725320114 0101	1.529824675655 5098	1.520419832119 5102	1.482128631716 18	1.503454115282 7499	1.504508750910 36	1.568040131331 2098	1.515820936628 58	1.508414271348 77
SCB	1.471757870370 74	1.442894128341 22	1.442650287212 96	1.559684469966 27	1.474384355846 12	1.436270111860 27	1.471730820193 45	1.443235178655 45	1.442407889790 74
SCC	1.106660856425 4499	1.097696491226 71	1.123468551081 95	1.137212373055 99	1.129000851546 81	1.130615387802 39	1.114012575310 43	1.099957616334 2402	1.122674888110 19
TASCO	2.593987195625 41	2.575404438151 37	2.415955752703 56	2.671659461851 43	2.653121116697 48	2.485417092309 91	2.611815096516 05	2.571972624322 47	2.416348597483 72
TCAP	1.168549363021 0602	1.147715156820 05	1.151521931556 66	1.189926257027 16	1.159291522299 4902	1.139197105258 28	1.169222267982 16	1.150705226166 5	1.151856684445 23
TMB	1.515034086322 6899	1.513433289462 52	1.421851120381 49	1.519414436728 88	1.509959760842 17	1.525916081160 53	1.545808723464 58	1.512355494577 3399	1.421760942570 0801
TOP	1.555589233862 02	1.552782872622 36	1.541139763803 48	1.559672794503 49	1.550016018190 18	1.569834247727 62	1.571076278733 8	1.555089500850 26	1.542438764597 52
TPIPL	1.817462014385 67	1.815393967037 9302	1.686244592268 1097	1.948756566739 96	1.955666352005 31	2.100541578327 91	1.842978815843 14	1.809707637544 88	1.685747715927 001
TRUE	2.058803709368 22	2.028285983626 8803	2.033119284165 41	2.022380369477 5997	2.010857735910 02	2.078124717480 4602	2.056272740081 4	2.031085795939 61	2.031291476517 11
TTW	0.856108853296 818	0.850523268164 296	0.828512871548 1869	0.914242215358 609	0.877891529887 26	0.848684461796 2661	0.863935737843 7299	0.849487178815 576	0.828715632321 261
TU	1.403001797359 69	1.366405556105 71	1.375508229602 83	1.437577274379 93	1.369683679126 19	1.332574486485 48	1.414596203746 2599	1.371563816811 4002	1.357945733426 96
WHA	1.761299561661 83	1.696322080039 2802	1.570737251771 20701	1.867557381357 61	1.867582940564 35	1.791503618980 8002	1.743932956685 0001	1.683625623714 64	1.568530264746 37

Direction Prediction Script

PredictDirection1 : Time Series

Table 52 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.7368421052632	43.1578947368421	43.1578947368421	44.7368421052632	43.1578947368421	43.684210526315795	43.1578947368421	43.1578947368421	43.1578947368421
AOT	60.0	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	60.52631578947369	61.052631578947405	61.052631578947405
BA	38.88888888888889	44.4444444444444	38.88888888888889	44.4444444444444	44.4444444444444	38.88888888888889	44.4444444444444	42.592592592592595	38.88888888888889
BANPU	41.57894736842211	41.57894736842211	47.8947368421053	41.57894736842211	41.57894736842211	45.78947368421105	41.57894736842211	41.57894736842211	50.52631578947737
BBL	44.7368421052632	44.7368421052632	55.2631578947368	44.7368421052632	44.7368421052632	53.15789473684211	44.7368421052632	44.7368421052632	54.73684210526319
BCP	42.105263157894704	48.421052631578895	41.57894736842211	46.842105263157904	48.947368421052526	44.210526315789	43.157894736842	49.473684210526	42.105263157894704
BDMS	62.1052631578947	62.6315789473684	62.6315789473684	62.6315789473684	62.6315789473684	62.63157894736842	61.578947368421105	62.6315789473684	62.6315789473684
BEC	45.26315789473368	45.26315789473368	49.473684210526	45.26315789473368	45.26315789473368	51.578947368421105	45.26315789473368	45.26315789473368	49.473684210526
BH	51.052631578947405	50.52631578947737	49.473684210526	51.052631578947405	50.52631578947737	49.473684210526	51.052631578947405	50.52631578947737	49.473684210526
BLA	50.0	50.0	48.421052631578895	50.0	50.52631578947737	48.421052631578895	50.0	50.0	48.947368421052526
BTS	44.2105263157895	45.78947368421105	44.7368421052632	44.7368421052632	45.78947368421105	51.578947368421105	45.26315789473368	45.78947368421105	45.78947368421105
CBG	68.0	68.0	68.0	70.0	68.0	68.0	68.0	68.0	68.0
CENTEL	57.894736842105296	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	57.368421052631604	58.421052631578995	57.894736842105296
CK	57.894736842105296	58.421052631578995	61.052631578947405	58.421052631578995	58.421052631578995	60.52631578947369	57.894736842105296	58.421052631578995	61.052631578947405
CPALL	54.21052631578895	54.21052631578895	44.7368421052632	54.21052631578895	54.21052631578895	46.31578947368442	54.21052631578895	54.21052631578895	44.21052631578895
CPF	52.10526315789747	51.578947368421105	54.21052631578895	51.578947368421105	51.578947368421105	54.21052631578895	52.631578947368396	51.578947368421105	53.157894736842
CPN	50.52631578947737	51.578947368421105	56.31578947368442	53.157894736842	52.10526315789747	51.578947368421	50.0	51.578947368421105	55.578947368421105
DELTA	53.157894736842	52.631578947368396	45.26315789473368	53.157894736842	53.684210526315	45.26315789473368	53.157894736842	52.631578947368396	45.26315789473368
DTAC	50.0	50.0	46.842105263157904	50.52631578947737	50.0	46.31578947368442	50.0	50.0	46.842105263157904
EGCO	61.052631578947405	60.0	61.052631578947405	60.526315789473 69	59.473684210526 3	61.052631578947405	60.526315789473 69	60.0	61.052631578947405
GLOW	48.421052631578895	44.21052631578895	50.52631578947737	47.8947368421053	42.10526315789474	48.947368421052526	48.421052631578895	42.105263157894704	50.0
GPSC	56.25	56.25	56.25	56.25	56.25	62.5	56.25	56.25	56.25
HMPRO	52.10526315789747	53.684210526315 8	63.157894736842 1	53.684210526315 8	52.631578947368396	63.157894736842 1	53.684210526315 8	53.684210526315 8	63.157894736842 1
INTUCH	42.105263157894704	41.05263157894 74	42.105263157894704	41.578947368421 1	40.526315789473 7	42.105263157894704	42.105263157894704	41.05263157894 74	42.105263157894704
IRPC	54.21052631578895	56.842105263157904	50.52631578947737	53.684210526315 8	56.842105263157904	54.21052631578895	54.21052631578895	56.842105263157904	50.52631578947737
IVL	46.842105263157904	47.8947368421053	51.052631578947405	47.8947368421053	48.421052631578895	47.8947368421053	46.842105263157904	48.421052631578895	50.52631578947737
KBANK	44.7368421052632	45.26315789473368	54.73684210526319	44.7368421052632	45.26315789473368	54.21052631578895	44.7368421052632	45.26315789473368	54.73684210526319
KCE	57.894736842105296	60.0	60.0	60.0	60.0	60.0	58.9473684210526	60.0	60.0
KTB	39.473684210263	39.473684210263	60.526315789473 69	39.473684210263	39.473684210263	60.526315789473 69	39.473684210263	39.473684210263	60.526315789473 69
LH	39.4736842105263	38.9473684210526	61.57894736842 11	38.9473684210526	38.9473684210526	61.57894736842 11	39.4736842105263	38.4210526315789	61.57894736842 11
MINT	57.368421052631604	57.894736842105296	57.894736842105296	57.368421052631604	57.368421052631604	57.894736842105296	57.894736842105296	57.894736842105296	57.894736842105296
MTLS	51.020408163265294	48.9795918367347	53.0612244897959	51.020408163265294	51.020408163265294	51.020408163265294	51.020408163265294	51.020408163265294	53.0612244897959
PS	35.2631578947368	36.31578947368442	63.684210526315 8	36.842105263157 9	38.421052631578 95	63.684210526315 8	38.421052631578 95	40.0	63.684210526315 8
PTT	43.684210526315795	44.2105263157895	56.31578947368442	44.2105263157895	43.157894736842 1	55.263157894736 8	43.684210526315795	42.631578947368396	54.73684210526319
PTTEP	47.36842105263 16	47.36842105263 16	50.0	47.36842105263 16	47.36842105263 16	51.052631578947405	47.36842105263 16	47.36842105263 16	50.0
PTTGC	46.31578947368442	46.842105263157904	46.842105263157904	46.31578947368442	46.842105263157904	46.842105263157904	46.31578947368442	46.842105263157904	46.842105263157904
ROBINS	55.78947368421105	56.31578947368442	62.1052631578947	55.78947368421105	56.31578947368442	62.1052631578947	56.31578947368442	56.31578947368442	62.1052631578947
SAWAD	46.666666666667	51.111111111111 09	41.111111111111	52.222222222222	48.888888888889	41.111111111111	48.888888888889	50.0	41.111111111111
SCB	40.52631578947737	41.05263157894 74	58.947368421052 6	40.526315789473 7	41.05263157894 74	56.31578947368442	40.0	41.05263157894 74	58.947368421052 6
SCC	50.0	47.8947368421053	46.31578947368442	50.0	47.36842105263 16	49.473684210526	50.0	47.8947368421053	44.736842105263 2
TASCO	52.631578947368396	53.684210526315 8	41.05263157894 74	53.157894736842 1	53.684210526315 8	41.05263157894 74	52.10526315789747	53.684210526315 8	41.05263157894 74
TCAP	55.78947368421105	57.368421052631604	61.57894736842 11	56.31578947368442	56.842105263157904	61.57894736842 11	55.78947368421105	57.368421052631604	61.57894736842 11
TMB	44.7368421052632	46.842105263157904	56.31578947368442	45.26315789473368	46.842105263157904	56.31578947368442	45.78947368421105	46.842105263157904	56.31578947368442
TOP	58.421052631578995	60.0	60.0	59.473684210526 3	60.0	59.473684210526 3	60.0	60.0	60.0
TPIPL	56.842105263157904	56.31578947368442	52.10526315789747	56.31578947368442	56.31578947368442	51.052631578947405	56.31578947368442	56.31578947368442	52.10526315789747
TRUE	46.842105263157904	45.78947368421105	54.736842105263 19	46.842105263157904	44.736842105263 2	55.78947368421105	46.842105263157904	45.78947368421105	55.263157894737 68
TTW	37.368421052631604	37.894736842105296	65.263157894736 8	38.421052631578 9	38.421052631578 9	65.263157894736 8	38.421052631578 9	38.421052631578 9	65.263157894736 8
TU	50.52631578947737	51.578947368421105	53.157894736842 1	51.052631578947405	51.052631578947405	53.157894736842 1	50.52631578947737	52.105263157894 7	53.157894736842 1
WHA	41.05263157894 74	43.157894736842 1	50.0	42.631578947368396	43.157894736842 1	48.421052631578895	42.105263157894704	43.157894736842 1	50.0

Table 53 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	45.26315789473368	56.31578947368442	42.631578947368396	44.7368421052632	56.31578947368442	43.684210526315795	44.7368421052632	56.31578947368442
AOT	60.526315789473 69	44.2105263157895	38.947368421052 6	38.947368421052 6	38.947368421052 6	61.052631578947405	61.052631578947405	45.78947368421105	38.947368421052 6
BA	44.4444444444444	42.592592592592595	38.88888888888889	42.592592592592595	38.88888888888889	61.11111111111111	42.592592592592595	40.7407407407407	38.88888888888889
BANPU	58.421052631578995	58.421052631578995	58.421052631578995	41.05263157894 74	58.421052631578995	41.57894736842211	58.421052631578995	58.421052631578995	58.421052631578995
BBL	43.15789473684211	44.7368421052632	49.473684210526 3	42.631578947368396	55.263157894736 8	55.263157894736 8	41.57894736842211	44.7368421052632	50.0
BCP	61.57894736842 11	61.57894736842 11	38.421052631578 9	38.421052631578 9	38.421052631578 9	37.368421052631604	61.57894736842 11	61.57894736842 11	38.421052631578 9
BDMS	58.421052631578995	62.631578947368 4	53.684210526315 8	37.368421052631604	37.368421052631604	37.368421052631604	55.78947368421105	62.631578947368 4	55.78947368421105
BEC	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319	50.52631578947737	54.73684210526319	54.73684210526319	54.73684210526319
BH	50.52631578947737	50.0	49.473684210526 3	50.0	49.473684210526 3	49.473684210526 3	50.52631578947737	50.52631578947737	50.52631578947737
BLA	51.052631578947405	54.21052631578895	47.8947368421053	48.421052631578895	45.78947368421105	45.78947368421105	44.7368421052632	54.21052631578895	48.421052631578895
BTS	59.473684210526 3	59.473684210526 3	59.473684210526 3	40.52631578947737	40.52631578947737	40.52631578947737	59.473684210526 3	59.473684210526 3	59.473684210526 3
CBG	32.0	32.0	68.0	32.0	32.0	32.0	32.0	32.0	68.0
CENTEL	47.36842105263 16	45.78947368421105	41.57894736842 11	41.57894736842 11	41.57894736842 11	41.57894736842 11	58.421052631578995	45.26315789473368	41.57894736842 11
CK	44.7368421052632	40.0	38.947368421052 6	42.631578947368396	39.473684210526 3	47.89473684211053	47.36842105263 16	40.52631578947737	38.947368421052 6
CPALL	43.15789473684211	43.15789473684211	43.15789473684211	43.15789473684211	43.15789473684211	56.842105263157904	43.15789473684211	43.15789473684211	43.15789473684211
CPF	56.31578947368442	56.31578947368442	56.31578947368442	56.31578947368442	56.31578947368442	43.684210526315795	56.31578947368442	56.31578947368442	56.31578947368442
CPN	61.57894736842 11	61.57894736842 11	38.421052631578 9	38.947368421052 6	40.0	38.421052631578 9	61.57894736842 11	61.57894736842 11	38.421052631578 9
DELTA	54.73684210526319	45.26315789473368	54.73684210526319	54.73684210526319	45.78947368421105	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319
DTAC	52.10526315789747	51.052631578947405	48.421052631578895	50.0	50.0	50.0	51.578947368421105	52.10526315789747	48.947368421052526
EGCO	45.78947368421105	41.05263157894 74	38.947368421052 6	46.842105263157904	61.052631578947405	61.052631578947405	46.31578947368442	40.52631578947737	38.947368421052 6
GLOW	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	42.105263157894704	41.57894736842211	58.421052631578995	43.157894736842 1	58.421052631578995
GPSC	68.75	43.75	43.75	43.75	43.75	43.75	62.5	56.25	43.75
HMPRO	63.684210526315 8	63.157894736842 1	36.842105263157 9	36.842105263157 9	36.842105263157 9	40.526315789473 7	62.631578947368 4	63.157894736842 1	36.842105263157 9
INTUCH	42.105263157894704	42.105263157894704	45.263157894736 8	42.105263157894704	42.105263157894704	57.894736842105296	42.105263157894704	42.105263157894704	46.842105263157904
IRPC	56.842105263157904	56.842105263157904	43.157894736842 1	47.8947368421053	47.368421052616	43.157894736842 1	56.842105263157904	56.842105263157904	43.157894736842 1
IVL	57.368421052631604	57.368421052631604	42.631578947368396	42.210526315789 5	41.578947368421 1	57.368421052631604	57.368421052631604	57.368421052631604	42.631578947368396
KBANK	54.73684210526319	40.52631578947737	54.73684210526319	40.526315789473 7	44.736842105263 2	45.263157894736 8	47.368421052616	43.684210526315795	54.73684210526319
KCE	60.0	43.157894736842 1	51.578947368421105	40.526315789473 7	40.0	40.0	60.0	60.0	51.578947368421105
KTB	41.05263157894 74	42.105263157894704	54.73684210526319	39.473684210526 3	41.578947368421 1	52.10526315789747	41.57894736842211	41.57894736842211	55.78947368421105
LH	45.78947368421105	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	61.57894736842 11	38.4210526315789	38.4210526315789
MINT	44.2105263157895	42.105263157894704	42.105263157894704	45.263157894736 8	43.684210526315795	42.105263157894704	44.2105263157895	42.105263157894704	42.105263157894704
MTLS	51.020408163265294	51.020408163265294	51.020408163265294	48.9795918367347	48.9795918367347	48.9795918367347	51.020408163265294	51.020408163265294	51.020408163265294
PS	41.57894736842211	37.368421052631604	36.842105263157 9	38.421052631578 95	36.842105263157 9	36.31578947368442	40.526315789473 7	37.368421052631604	36.31578947368442
PTT	58.94736842105256	58.94736842105256	53.157894736842 1	41.052631578947 4	41.052631578947 4	51.578947368421105	58.94736842105256	58.94736842105256	52.10526315789747
PTTEP	52.631578947368396	52.631578947368396	52.631578947368396	53.684210526315 8	52.631578947368396	47.36842105263 16	52.631578947368396	52.631578947368396	52.631578947368396
PTTGC	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1
ROBINS	62.1052631578947	62.1052631578947	62.1052631578947	37.894736842105296	61.052631578947405	62.1052631578947	62.1052631578947	62.1052631578947	62.1052631578947
SAWAD	41.111111111111 1	41.111111111111 1	58.888888888889	41.111111111111 1	41.111111111111 1	58.888888888889	41.111111111111 1	58.888888888889	58.888888888889
SCB	58.94736842105256	58.94736842105256	58.94736842105256	42.631578947368396	40.0	39.473684210526 3	58.94736842105256	58.94736842105256	58.94736842105256
SCC	43.15789473684211	41.57894736842211	60.0	42.105263157894704	41.578947368421 1	60.0	44.2105263157895	41.57894736842211	60.0
TASCO	46.31578947368442	46.31578947368442	46.31578947368442	46.31578947368442	46.31578947368442	46.31578947368442	46.31578947368442	46.31578947368442	46.31578947368442
TCAP	45.78947368421105	45.26315789473368	38.421052631578 9	43.15789473684211	41.578947368421 1	54.21052631578895	45.26315789473368	43.15789473684211	38.421052631578 9
TMB	45.78947368421105	56.31578947368442	49.473684210526 3	46.842105263157904	48.421052631578895	56.31578947368442	48.9473684210526	56.31578947368442	50.0
TOP	60.0	45.78947368421105	40.0	60.0	44.736842105263 2	40.0	60.0	45.26315789473368	40.0
TPIPL	55.78947368421105	55.78947368421105	55.78947368421105	42.631578947368396	44.210526315789 5	55.78947368421105	55.78947368421105	55.78947368421105	55.78947368421105
TRUE	54.73684210526319	54.73684210526319	54.73684210526319	39.473684210526 3	45.78947368421105	41.052631578947 4	54.73684210526319	54.73684210526319	54.73684210526319
TTW	65.263157894736 8	65.263157894736 8	64.736842105263 21	35.789473684210 5	65.263157894736 8	65.263157894736 8	65.263157894736 8	65.263157894736 8	65.263157894736 8
TU	60.0	60.0	60.0	60.0	60.0	42.631578947368396	60.0	60.0	60.0
WHA	55.263157894736 8	55.263157894736 8	55.263157894736 8	44.736842105263 2	44.736842105263 2	55.263157894736 8	55.263157894736 8	55.263157894736 8	55.263157894736 8

Table 54 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	44.2105263157895	56.3157894736842	42.631578947368396	43.15789473684421	56.31578947368421	43.684210526315795	44.2105263157895	56.3157894736842
AOT	61.052631578947405	61.052631578947405	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	61.052631578947405	61.052631578947405	38.9473684210526
BA	44.44444444444444	42.592592592592595	38.88888888888889	44.44444444444444	38.88888888888889	61.11111111111111	44.44444444444444	44.44444444444444	38.88888888888889
BANPU	58.421052631578995	58.421052631578995	58.421052631578995	40.0	58.421052631578995	41.57894736842211	58.421052631578995	58.421052631578995	58.421052631578995
BBL	42.631578947368396	44.7368421052632	48.9473684210526	43.684210526315795	55.263157894736	47.89473684211053	45.263157894736	45.263157894736	49.4736842105263
BCP	61.57894736842211	61.57894736842211	38.4210526315789	38.4210526315789	38.4210526315789	37.368421052631604	61.57894736842211	61.57894736842211	38.4210526315789
BDMS	58.421052631578995	62.6315789473684	53.68421052631579	37.368421052631604	37.368421052631604	37.368421052631604	58.421052631578995	62.6315789473684	55.7894736842105
BEC	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319	50.5263157894737	54.73684210526319	54.73684210526319	54.73684210526319
BH	50.5263157894737	51.052631578947405	48.421052631578895	51.578947368421105	49.473684210526	49.473684210526	50.5263157894737	50.5263157894737	48.421052631578895
BLA	45.7894736842105	54.2105263157895	48.421052631578895	45.7894736842105	45.7894736842105	45.7894736842105	48.421052631578895	54.2105263157895	48.421052631578895
BTS	59.4736842105263	59.4736842105263	59.4736842105263	40.5263157894737	40.5263157894737	40.5263157894737	59.4736842105263	59.4736842105263	59.4736842105263
CBG	32.0	32.0	68.0	50.0	52.0	32.0	32.0	32.0	68.0
CENTEL	47.36842105263 16	45.2631578947368	41.5789473684211	41.5789473684211	41.5789473684211	41.5789473684211	58.421052631578995	45.2631578947368	41.5789473684211
CK	47.89473684210053	38.9473684210526	38.9473684210526	47.368421052631 6	47.89473684210053	61.052631578947405	47.368421052631 6	40.0	38.9473684210526
CPALL	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	56.842105263157904	43.1578947368421	43.1578947368421	43.1578947368421
CPF	56.3157894736842	56.3157894736842	56.3157894736842	56.3157894736842	56.3157894736842	56.3157894736842	56.3157894736842	56.3157894736842	56.3157894736842
CPN	61.57894736842211	61.57894736842211	38.4210526315789	38.4210526315789	40.5263157894737	38.4210526315789	61.57894736842211	61.57894736842211	38.4210526315789
DELTA	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319	44.7368421052632	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319
DTAC	52.1052631578947	51.052631578947405	45.2631578947368	50.0	52.1052631578947	50.0	53.15789473684421	52.1052631578947	45.7894736842105
EGCO	46.31578947368421	39.4736842105263	38.9473684210526	46.842105263157904	61.052631578947405	61.052631578947405	47.36842105263 16	38.9473684210526	38.9473684210526
GLOW	58.421052631578995	43.157894736842 1	58.421052631578995	41.57894736842211	39.473684210526 3	41.57894736842211	58.421052631578995	49.473684210526 3	58.421052631578995
GPSC	62.5	43.75	43.75	43.75	43.75	43.75	50.0	56.25	43.75
HMPRO	62.631578947368 4	63.157894736842 1	36.842105263157 9	36.842105263157 9	35.789473684210 5	40.526315789473 7	64.210526315789 5	63.157894736842 1	36.842105263157 9
INTUCH	42.105263157894704	42.105263157894704	45.2631578947368	42.631578947368396	43.684210526315795	57.894736842105296	42.105263157894704	42.105263157894704	46.842105263157904
IRPC	56.842105263157904	56.842105263157904	43.157894736842 1	47.89473684210053	45.263157894736 8	43.157894736842 1	56.842105263157904	56.842105263157904	43.157894736842 1
IVL	57.368421052631604	57.368421052631604	42.631578947368396	43.684210526315795	42.631578947368396	42.631578947368396	57.368421052631604	57.368421052631604	42.631578947368396
KBANK	54.73684210526319	43.684210526315795	50.0	40.0	42.631578947368396	38.9473684210526	41.57894736842211	43.684210526315795	51.578947368421105
KCE	60.0	60.0	51.052631578947405	43.157894736842 1	40.0	40.0	60.0	60.0	51.578947368421105
KTB	41.05263157894 74	38.9473684210526	54.736842105263 19	39.473684210526 3	41.05263157894 74	52.1052631578947	42.631578947368396	42.105263157894704	55.789473684210 5
LH	61.57894736842211	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	61.57894736842211	38.4210526315789	38.4210526315789
MINT	43.684210526315795	43.684210526315795	42.105263157894704	42.105263157894704	43.684210526315795	42.105263157894704	43.684210526315795	43.684210526315795	42.105263157894704
MTLS	51.020408163265294	51.020408163265294	51.020408163265294	48.9795918367347	51.020408163265294	48.9795918367347	51.020408163265294	51.020408163265294	51.020408163265294
PS	43.684210526315795	37.894736842105296	36.3157894736842	36.3157894736842	36.3157894736842	36.8421052631579	40.5263157894737	36.8421052631579	36.3157894736842
PTT	58.9473684210526	58.9473684210526	53.1578947368421	41.05263157894 74	41.05263157894 74	51.578947368421105	58.9473684210526	58.9473684210526	52.1052631578947
PTTEP	52.631578947368396	52.631578947368396	52.631578947368396	58.9473684210526	51.052631578947405	52.631578947368396	52.631578947368396	52.631578947368396	52.631578947368396
PTTGC	53.1578947368421	53.1578947368421	53.1578947368421	53.1578947368421	53.1578947368421	53.1578947368421	53.1578947368421	53.1578947368421	53.1578947368421
ROBINS	62.1052631578947	62.1052631578947	62.1052631578947	37.894736842105296	37.894736842105296	62.1052631578947	62.1052631578947	62.1052631578947	62.1052631578947
SAWAD	41.11111111111111	41.11111111111111	58.88888888888889	41.11111111111111	41.11111111111111	58.88888888888889	41.11111111111111	41.11111111111111	58.88888888888889
SCB	58.9473684210526	58.9473684210526	58.9473684210526	42.631578947368396	40.0	39.4736842105263	58.9473684210526	58.9473684210526	58.9473684210526
SCC	44.21052631578995	42.105263157894704	60.0	41.57894736842211	42.105263157894704	60.0	43.684210526315795	42.105263157894704	60.0
TASCO	46.31578947368421	46.31578947368421	46.3157894736842	43.1578947368421	46.31578947368421	46.3157894736842	46.31578947368421	46.31578947368421	46.31578947368421
TCAP	46.31578947368421	45.7894736842105	38.4210526315789	45.263157894736 8	41.57894736842211	54.21052631578995	47.36842105263 16	43.157894736842 1	38.4210526315789
TMB	49.4736842105263	45.7894736842105	49.4736842105263	45.7894736842105	44.7368421052632	56.3157894736842	47.89473684210053	45.263157894736 8	50.0
TOP	60.0	44.7368421052632	40.0	47.89473684210053	43.157894736842 1	40.0	60.0	44.2105263157895	40.0
TPIPL	55.7894736842105	55.7894736842105	46.3157894736842	44.7368421052632	44.7368421052632	55.7894736842105	55.7894736842105	55.7894736842105	46.31578947368421
TRUE	54.73684210526319	54.73684210526319	54.73684210526319	41.05263157894 74	41.57894736842211	54.73684210526319	54.73684210526319	54.73684210526319	54.73684210526319
TTW	65.2631578947368	65.2631578947368	61.57894736842211	34.7368421052632	34.7368421052632	34.7368421052632	65.2631578947368	65.2631578947368	64.2105263157895
TU	60.0	60.0	60.0	60.0	60.0	42.631578947368396	60.0	60.0	60.0
WHA	55.2631578947368	55.2631578947368	55.2631578947368	44.7368421052632	44.7368421052632	55.2631578947368	55.2631578947368	55.2631578947368	55.2631578947368

Table 55 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	43.684210526315795	42.631578947368396	44.7368421052632	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795
AOT	38.9473684210526	42.631578947368396	38.9473684210526	61.052631578947405	44.2105263157895	56.3157894736842	38.9473684210526	43.157894736842 1	38.9473684210526
BA	38.88888888888889	38.88888888888889	38.88888888888889	61.11111111111111	46.296296296296305	55.55555555555556	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	37.368421052631604	41.57894736842211	41.57894736842211	58.421052631578995	40.5263157894737	41.57894736842211	38.9473684210526	40.0	41.57894736842211
BBL	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632
BCP	38.4210526315789	41.0526315789474	38.4210526315789	61.57894736842211	42.631578947368396	38.4210526315789	38.4210526315789	40.5263157894737	38.4210526315789
BDMS	37.368421052631604	37.368421052631604	37.368421052631604	62.6315789473684	43.684210526315795	62.6315789473684	37.368421052631604	37.368421052631604	37.368421052631604
BEC	45.2631578947368	48.421052631578895	45.2631578947368	54.73684210526319	51.052631578947405	53.68421052631 58	48.421052631578895	49.473684210526 3	45.2631578947368
BH	52.1052631578947	50.0	49.473684210526 3	50.5263157894737	51.578947368421105	49.473684210526 3	51.052631578947405	50.5263157894737	49.4736842105263
BLA	45.7894736842105	45.7894736842105	45.7894736842105	51.052631578947405	45.7894736842105	47.89473684210053	45.7894736842105	45.7894736842105	45.7894736842105
BTS	43.1578947368421	41.57894736842211	40.5263157894737	59.473684210526 3	42.105263157894704	40.5263157894737	44.2105263157895	42.105263157894704	40.5263157894737
CBG	32.0	32.0	32.0	50.0	32.0	68.0	32.0	32.0	32.0
CENTEL	45.7894736842105	42.631578947368396	41.5789473684211	57.894736842105296	45.7894736842105	41.5789473684211	44.7368421052632	43.684210526315795	41.5789473684211
CK	38.9473684210526	38.9473684210526	38.9473684210526	61.052631578947405	38.9473684210526	61.052631578947405	38.9473684210526	38.9473684210526	38.9473684210526
CPALL	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421
CPF	44.7368421052632	44.7368421052632	43.684210526315795	56.3157894736842	45.2631578947368	43.684210526315795	44.7368421052632	45.2631578947368	43.684210526315795
CPN	38.9473684210526	41.57894736842211	38.4210526315789	61.052631578947405	38.9473684210526	55.7894736842105	38.9473684210526	39.473684210526 3	38.4210526315789
DELTA	45.7894736842105	44.2105263157895	45.2631578947368	54.73684210526319	45.2631578947368	45.2631578947368	46.842105263157904	44.7368421052632	45.2631578947368
DTAC	52.1052631578947	50.0	49.473684210526 3	47.89473684210053	50.0	48.421052631578895	53.15789473684421	50.0	47.89473684210053
EGCO	38.9473684210526	38.9473684210526	38.9473684210526	61.052631578947405	38.4210526315789	47.89473684210053	38.9473684210526	38.9473684210526	38.9473684210526
GLOW	41.57894736842211	41.052631578947 4	41.57894736842211	60.0	45.2631578947368	55.7894736842105	41.57894736842211	41.052631578947 4	41.57894736842211
GPSC	43.75	43.75	43.75	62.5	43.75	43.75	43.75	43.75	43.75
HMPRO	36.842105263157 9	36.842105263157 9	36.842105263157 9	63.157894736842 1	36.842105263157 9	36.842105263157 9	36.842105263157 9	36.842105263157 9	36.842105263157 9
INTUCH	42.105263157894704	42.105263157894704	42.105263157894704	57.894736842105296	43.684210526315795	42.105263157894704	42.105263157894704	42.105263157894704	42.105263157894704
IRPC	43.15789473684421	43.15789473684421	43.15789473684421	56.842105263157904	43.15789473684421	43.15789473684421	43.15789473684421	43.15789473684421	43.15789473684421
IVL	42.631578947368396	42.105263157894704	42.631578947368396	44.7368421052632	42.105263157894704	42.631578947368396	41.57894736842211	41.052631578947 4	42.631578947368396
KBANK	40.5263157894737	44.2105263157895	45.2631578947368	54.73684210526319	45.7894736842105	45.2631578947368	40.0	43.684210526315795	45.2631578947368
KCE	40.0	43.684210526315795	40.0	60.0	42.631578947368396	60.0	40.0	41.57894736842211	40.0
KTB	42.105263157894704	38.9473684210526	39.473684210526 3	42.105263157894704	41.052631578947 4	39.4736842105263	40.5263157894737	39.473684210526 3	39.473684210526 3
LH	38.4210526315789	38.4210526315789	38.4210526315789	61.57894736842211	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789
MINT	42.105263157894704	42.105263157894704	42.105263157894704	57.894736842105296	44.7368421052632	53.68421052631 58	42.105263157894704	42.105263157894704	42.105263157894704
MTLS	57.142857142857096	53.06122448979591	48.9795918367347	44.8979591836735	51.020408163265294	48.9795918367347	55.102040816326 5	51.020408163265294	48.9795918367347
PS	36.3157894736842	37.368421052631604	36.3157894736842	63.68421052631 58	37.368421052631604	36.3157894736842	36.3157894736842	37.368421052631604	36.3157894736842
PTT	41.05263157894 74	41.05263157894 74	58.9473684210526	58.9473684210526	39.473684210526 3	41.05263157894 74	41.05263157894 74	41.05263157894 74	58.9473684210526
PTTEP	52.631578947368396	43.684210526315795	52.631578947368396	52.631578947368396	52.631578947368396	47.36842105263 16	52.631578947368396	42.631578947368396	52.631578947368396
PTTGC	43.1578947368421	47.36842105263 16	46.842105263157904	52.631578947368396	46.842105263157904	52.1052631578947	45.2631578947368	47.36842105263 16	46.842105263157904
ROBINS	37.894736842105296	37.894736842105296	37.894736842105296	38.4210526315789	38.4210526315789	62.1052631578947	37.894736842105296	38.4210526315789	37.894736842105296
SAWAD	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111
SCB	42.105263157894704	39.4736842105263	58.9473684210526	58.9473684210526	39.4736842105263	41.0526315789474	41.0526315789474	39.4736842105263	58.9473684210526
SCC	43.684210526315795	40.5263157894737	40.0	60.0	41.57894736842211	40.0	42.105263157894704	40.0	40.0
TASCO	46.31578947368421	46.31578947368421	46.3157894736842	53.6842105263158	47.36842105263 16	55.7894736842105	46.31578947368421	46.31578947368421	46.31578947368421
TCAP	38.4210526315789	40.5263157894737	38.4210526315789	61.57894736842211	42.105263157894704	38.4210526315789	38.4210526315789	41.57894736842211	38.4210526315789
TMB	43.684210526315795	43.684210526315795	43.684210526315795	54.73684210526319	48.421052631578895	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795
TOP	44.7368421052632	41.0526315789474	40.0	60.0	44.2105263157895	40.0	44.7368421052632	40.5263157894737	40.0
TPIPL	44.2105263157895	44.2105263157895	44.2105263157895	55.7894736842105	42.105263157894704	45.7894736842105	44.2105263157895	44.2105263157895	44.2105263157895
TRUE	45.2631578947368	45.2631578947368	45.2631578947368	54.73684210526319	44.7368421052632	53.68421052631 58	45.2631578947368	45.2631578947368	45.2631578947368
TTW	34.7368421052632	34.7368421052632	34.7368421052632	45.2631578947368	36.3157894736842	34.7368421052632	34.7368421052632	34.7368421052632	34.7368421052632
TU	44.7368421052632	40.5263157894737	40.0	60.0	42.631578947368396	54.73684210526319	45.2631578947368	41.052631578947 4	40.0
WHA	44.7368421052632	46.842105263157904	44.7368421052632	53.1578947368421	44.7368421052632	55.7894736842105	44.7368421052632	46.842105263157904	44.7368421052632

PredictDirection2 : Historical data

Table 56 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.2105263157895	43.684210526315795	44.7368421052632	43.684210526315795	44.7368421052632	43.15789473684211	44.2105263157895	44.7368421052632	44.7368421052632
AOT	57.368421052631604	57.894736842105296	59.4736842105263	57.894736842105296	56.842105263157904	60.0	58.421052631578995	58.421052631578995	58.94736842105263
BA	59.2592592592593	55.55555555555556	55.55555555555556	61.111111111111114	59.2592592592593	55.55555555555556	51.851851851851805	51.851851851851805	55.55555555555556
BANPU	43.684210526315795	40.52631578947368	42.631578947368396	43.684210526315795	40.52631578947368	42.631578947368396	41.0526315789474	41.0526315789474	42.631578947368396
BBL	42.631578947368396	42.105263157894704	42.631578947368396	41.5789473684211	42.105263157894704	42.105263157894704	42.105263157894704	41.0526315789474	42.631578947368396
BCP	51.578947368421105	52.631578947368396	55.263157894368	52.1052631578947	51.578947368421105	50.52631578947368	52.631578947368396	53.157894736842	55.263157894368
BDMS	57.894736842105296	57.894736842105296	56.842105263157904	57.368421052631604	57.894736842105296	56.842105263157904	57.894736842105296	57.368421052631604	56.842105263157904
BEC	46.315789473684	45.26315789473368	45.789473684210	46.315789473684	45.26315789473368	46.315789473684	46.315789473684	45.26315789473368	45.789473684210
BH	52.10526315789474	56.842105263157904	55.263157894368	54.2105263157895	53.684210526315	54.736842105263	54.736842105263	53.157894736842	55.263157894368
BLA	46.842105263157904	50.0	50.0	46.842105263157904	49.473684210526	49.473684210263	47.368421052636	49.473684210526	50.0
BTS	53.684210526315	50.0	55.78947368421	48.421052631578895	49.473684210526	53.684210526315	52.10526315789474	50.0	55.78947368421
CBG	70.0	70.0	68.0	70.0	70.0	70.0	70.0	70.0	68.0
CENTEL	54.736842105263	57.368421052631604	57.894736842105296	54.736842105263	56.842105263157904	56.842105263157904	54.736842105263	56.842105263157904	57.894736842105296
CK	52.631578947368396	56.31578947368421	54.2105263157895	54.736842105263	53.157894736842	53.684210526315	52.10526315789474	55.78947368421	54.2105263157895
CPALL	53.684210526315	53.684210526315	56.842105263157904	54.2105263157895	54.2105263157895	56.842105263157904	54.2105263157895	53.684210526315	56.31578947368421
CPF	54.736842105263	52.10526315789474	46.3157894736842	49.473684210526	51.052631578947405	46.842105263157904	51.052631578947405	50.0	46.31578947368421
CPN	57.368421052631604	56.842105263157904	56.842105263157904	57.368421052631604	56.842105263157904	57.368421052631604	56.842105263157904	57.368421052631604	56.842105263157904
DELTA	55.26315789473368	52.10526315789474	52.631578947368396	53.6842105263158	52.1052631578947	55.26315789473368	55.78947368421105	51.578947368421105	52.631578947368396
DTAC	49.473684210263	50.0	48.9473684210526	50.0	50.0	48.9473684210526	49.473684210526	50.0	48.9473684210526
EGCO	54.736842105263	55.26315789473368	55.263157894368	52.631578947368396	56.3157894736842	54.736842105263	55.26315789473368	55.26315789473368	55.26315789473368
GLOW	49.473684210263	48.9473684210526	50.0	47.8947368421053	48.9473684210526	50.0	49.473684210263	48.9473684210526	50.0
GPSC	56.25	56.25	62.5	56.25	56.25	62.5	56.25	62.5	62.5
HMPRO	55.78947368421105	57.894736842105296	58.421052631578995	57.894736842105296	59.473684210263	57.368421052631604	55.78947368421105	57.894736842105296	58.94736842105263
INTUCH	41.0526315789474	41.0526315789474	40.52631578947368	41.0526315789474	40.0	42.105263157894704	41.0526315789474	41.0526315789474	40.52631578947368
IRPC	54.2105263157895	54.736842105263	54.736842105263	55.78947368421105	54.736842105263	55.26315789473368	53.684210526315	54.736842105263	54.2105263157895
IVL	56.842105263157904	58.421052631578995	55.263157894368	55.78947368421105	56.842105263157904	55.78947368421105	58.94736842105263	56.842105263157904	55.26315789473368
KBANK	43.157894736842	44.21052631578947	43.684210526315795	43.157894736842	45.263157894368	45.26315789473368	41.57894736842105	46.315789473684	43.157894736842
KCE	59.473684210263	60.0	60.52631578947369	58.421052631578995	60.0	60.52631578947369	58.94736842105263	61.052631578947405	60.52631578947369
KTB	50.0	45.26315789473368	55.263157894368	50.0	45.263157894368	51.578947368421105	50.0	46.315789473684	55.263157894368
LH	42.105263157894704	43.684210526315795	48.421052631578895	46.315789473684	44.2105263157895	48.421052631578895	41.5789473684211	43.684210526315795	48.94736842105263
MINT	58.94736842105263	59.473684210263	57.894736842105296	58.421052631578995	59.473684210263	57.894736842105296	59.473684210263	59.473684210263	57.894736842105296
MTLS	51.020408163265294	55.1020408163265	51.020408163265294	53.0612244897959	51.020408163265294	48.9795918367347	53.0612244897959	55.1020408163265	51.020408163265294
PS	58.94736842105263	53.684210526315	51.052631578947405	54.2105263157895	53.684210526315	49.473684210263	59.473684210263	48.421052631578895	50.526315789474
PTT	45.263157894736842	42.631578947368396	41.0526315789474	41.0526315789474	42.631578947368396	40.526315789474737	47.368421052636	42.631578947368396	41.0526315789474
PTTEP	48.421052631578895	46.315789473684	48.9473684210526	47.8947368421053	45.263157894736842	51.052631578947405	47.368421052636	45.789473684210	48.94736842105263
PTTGC	53.157894736842	54.736842105263	58.94736842105263	54.2105263157895	55.263157894736842	55.78947368421105	54.2105263157895	54.2105263157895	58.421052631578995
ROBINS	51.578947368421105	52.631578947368396	51.578947368421105	53.157894736842	53.157894736842	51.052631578947405	51.052631578947405	50.526315789474737	51.578947368421105
SAWAD	36.66666666666667	38.888888888888889	44.444444444444	42.2222222222222	40.0	45.555555555556	41.111111111111	38.888888888888889	44.444444444444
SCB	42.105263157894704	41.57894736842105	44.2105263157895	40.526315789473737	42.105263157894704	43.684210526315795	41.0526315789474	42.105263157894704	44.2105263157895
SCC	46.842105263157904	43.684210526315795	49.473684210263	44.2105263157895	43.157894736842	46.315789473684	45.263157894736842	43.684210526315795	49.473684210263
TASCO	54.2105263157895	53.684210526315	51.578947368421105	53.157894736842	53.684210526315	53.684210526315	54.2105263157895	53.684210526315	51.578947368421105
TCAP	55.263157894736842	54.2105263157895	56.315789473684	54.736842105263	54.736842105263	53.157894736842	55.78947368421105	54.2105263157895	56.315789473684
TMB	54.736842105263	52.631578947368396	56.842105263157904	56.842105263157904	53.157894736842	54.2105263157895	54.2105263157895	54.736842105263	56.842105263157904
TOP	59.473684210263	60.0	60.0	59.473684210263	59.473684210263	60.0	59.473684210263	59.473684210263	60.0
TPIPL	47.8947368421053	46.842105263157904	50.526315789473737	48.421052631578895	50.0	50.0	47.368421052636	48.9473684210526	52.10526315789474
TRUE	44.736842105263	45.789473684210	54.736842105263	45.789473684210	46.842105263157904	52.631578947368396	45.789473684210	45.263157894736842	54.736842105263
TTW	44.2105263157895	38.42105263157895	50.0	41.0526315789474	41.578947368421105	48.9473684210526	44.2105263157895	39.473684210526	49.473684210263
TU	54.2105263157895	54.2105263157895	55.263157894736842	50.526315789473737	53.157894736842	54.736842105263	55.78947368421105	53.684210526315	55.263157894736842
WHA	57.894736842105296	54.736842105263	56.315789473684	58.94736842105263	57.368421052631604	55.78947368421105	58.421052631578995	54.2105263157895	56.315789473684

Table 57 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.105263157894704	40.52631578947368	44.7368421052632	43.684210526315795	42.631578947368396	46.315789473684	41.5789473684211	42.631578947368396	44.2105263157895
AOT	50.0	50.526315789473737	48.421052631578895	49.473684210263	53.157894736842	47.368421052636	50.0	50.526315789473737	48.421052631578895
BA	50.0	48.148148148148	55.55555555555556	59.2592592592593	61.111111111111114	61.111111111111114	55.55555555555556	48.148148148148	55.55555555555556
BANPU	50.526315789473737	49.473684210263	51.052631578947405	44.2105263157895	42.105263157894704	46.315789473684	51.052631578947405	50.0	51.052631578947405
BBL	54.2105263157895	52.10526315789474	45.263157894368	52.1052631578947	47.894736842105263	48.421052631578895	54.2105263157895	51.578947368421105	45.263157894736842
BCP	47.368421052636	44.21052631578947	50.0	43.157894736842	45.263157894736842	48.421052631578895	47.894736842105263	44.736842105263	50.0
BDMS	49.473684210263	49.473684210263	52.631578947368396	47.894736842105263	49.473684210263	49.473684210263	51.578947368421105	47.894736842105263	52.10526315789474
BEC	58.421052631578995	55.78947368421105	53.157894736842	53.157894736842	49.473684210263	51.578947368421105	57.894736842105296	55.78947368421105	53.684210526315
BH	52.631578947368396	52.631578947368396	48.947368421052636	50.526315789473737	47.894736842105263	51.578947368421105	52.10526315789474	53.157894736842	48.947368421052636
BLA	48.421052631578895	48.947368421052636	49.473684210263	44.736842105263	50.0	48.421052631578895	50.0	48.421052631578895	49.473684210263
BTS	53.157894736842	50.0	48.947368421052636	44.736842105263	41.052631578947	46.315789473684	53.157894736842	50.526315789473737	48.947368421052636
CBG	38.0	44.0	52.0	56.00000000000001	52.0	54.0	38.0	44.0	54.0
CENTEL	38.9473684210526	45.789473684210	47.368421052636	47.8947368421053	48.421052631578895	50.526315789473737	39.473684210263	46.315789473684	47.368421052636
CK	47.368421052636	42.631578947368396	42.105263157894704	53.157894736842	52.631578947368396	48.421052631578895	46.315789473684	42.631578947368396	42.105263157894704
CPALL	45.263157894736842	44.21052631578947	45.789473684210	43.157894736842	43.157894736842	46.842105263157904	45.263157894736842	46.315789473684	46.315789473684
CPF	53.157894736842	53.684210526315	53.157894736842	51.052631578947405	51.052631578947405	48.421052631578895	53.157894736842	52.631578947368396	53.157894736842
CPN	51.578947368421105	52.10526315789474	56.842105263157904	43.684210526315795	41.052631578947	44.21052631578947	52.10526315789474	52.10526315789474	56.842105263157904
DELTA	44.736842105263	46.842105263157904	45.789473684210	48.421052631578895	46.842105263157904	48.947368421052636	43.684210526315795	46.315789473684	45.789473684210
DTAC	51.052631578947405	52.631578947368396	54.2105263157895	51.052631578947405	56.842105263157904	55.789473684210	52.631578947368396	53.157894736842	53.684210526315
EGCO	46.842105263157904	46.315789473684	51.578947368421105	39.473684210263	41.052631578947	40.526315789473737	46.315789473684	47.368421052636	51.052631578947405
GLOW	47.368421052636	46.315789473684	48.421052631578895	43.684210526315795	42.631578947368396	45.263157894736842	48.947368421052636	46.315789473684	48.421052631578895
GPSC	37.5	56.25	50.0	43.75	43.75	43.75	43.75	62.5	50.0
HMPRO	53.157894736842	55.263157894736842	52.631578947368396	46.315789473684	50.0	50.526315789473737	53.684210526315	54.736842105263	52.631578947368396
INTUCH	44.736842105263	42.631578947368396	47.8947368421053	50.526315789473737	48.947368421052636	52.10526315789474	44.2105263157895	45.263157894736842	47.8947368421053
IRPC	40.526315789473737	42.631578947368396	47.894736842105263	43.157894736842	41.578947368421	46.315789473684	44.2105263157895	42.105263157894704	47.894736842105263
IVL	52.10526315789474	55.789473684210	57.894736842105296	54.2105263157895	52.105263157894704	52.105263157894704	52.10526315789474	55.789473684210	58.421052631578995
KBANK	52.10526315789474	48.421052631578895	49.473684210263	51.052631578947405	44.736842105263	46.842105263157904	53.157894736842	48.947368421052636	49.473684210263
KCE	51.052631578947405	51.052631578947405	50.526315789473737	42.631578947368396	41.052631578947	43.157894736842	50.526315789473737	50.526315789473737	51.052631578947405
KTB	43.157894736842	40.0	37.894736842105296	43.157894736842	42.105263157894704	45.789473684210	42.105263157894704	38.947368421052636	37.894736842105296
LH	48.421052631578895	50.526315789473737	51.578947368421105	46.842105263157904	43.684210526315795	42.105263157894704	47.368421052636	48.947368421052636	52.10526315789474
MINT	52.631578947368396	49.473684210263	53.684210526315	54.2105263157895	48.421052631578895	43.684210526315795	53.157894736842	48.421052631578895	53.684210526315
MTLS	55.1020408163265	51.020408163265294	51.020408163265294	53.0612244897959	46.9387755102041	48.9795918367347	53.0612244897959	53.0612244897959	51.020408163265294
PS	41.578947368421	45.263157894736842	50.526315789473737	45.263157894736842	48.947368421052636	45.789473684210	41.578947368421	44.736842105263	50.526315789473737
PTT	51.578947368421105	51.052631578947405	53.684210526315	45.789473684210	47.368421052636	47.894736842105263	51.052631578947405	51.052631578947405	54.2105263157895
PTTEP	55.263157894736842	58.421052631578995	49.473684210263	57.368421052631604	55.263157894736842	47.894736842105263	55.78947368421105	58.947368421052636	50.0
PTTGC	53.157894736842	50.526315789473737	48.947368421052636	53.157894736842	50.526315789473737	52.631578947368396	53.157894736842	50.526315789473737	48.947368421052636
ROBINS	51.052631578947405	50.0	52.631578947368396	53.684210526315	54.2105263157895	51.578947368421105	50.0	49.473684210263	52.631578947368396
SAWAD	43.333333333333	45.555555555556	42.2222222222222	56.6666666666667	48.888888888889	53.333333333333	43.333333333333	45.555555555556	43.333333333333
SCB	50.526315789473737	52.10526315789474	49.473684210263	50.0	48.421052631578895	47.894736842105263	52.10526315789474	52.10526315789474	50.0
SCC	48.947368421052636	46.315789473684	50.0	46.842105263157904	50.0	45.789473684210	50.0	46.315789473684	50.0
TASCO	45.789473684210	45.263157894736842	47.368421052636	46.315789473684	46.315789473684	49.473684210263	46.315789473684	45.263157894736842	46.842105263157904
TCAP	53.684210526315	52.10526315789474	52.631578947368396	48.421052631578895	51.052631578947405	52.631578947368396	54.2105263157895	52.631578947368396	53.157894736842
TMB	52.631578947368396	48.421052631578895	49.473684210263	44.736842105263	49.473684210263	46.842105263157904	51.578947368421105	49.473684210263	49.473684210263
TOP	50.0	46.315789473684	49.473684210263	53.157894736842	49.473684210263	51.578947368421105	50.526315789473737	46.842105263157904	49.473684210263
TPIPL	53.684210526315	53.684210526315	54.736842105263	51.578947368421105	53.684210526315	51.052631578947405	54.2105263157895	52.10526315789474	55.263157894736842
TRUE	48.421052631578895	43.157894736842	45.789473684210	47.368421052636	46.842105263157904	46.842105263157904	50.526315789473737	43.157894736842	45.789473684210
TTW	49.473684210263	48.421052631578895	42.631578947368396	44.2105263157895	44.2105263157895	40.526315789473737	46.315789473684	48.421052631578895	42.631578947368396
TU	46.842105263157904	42.105263157894704	47.368421052636	46.315789473684	47.368421052636	42.631578947368396	47.894736842105263	42.105263157894704	46.842105263157904
WHA	52.10526315789474	46.842105263157904	45.789473684210	48.947368421052636	46.315789473684	47.894736842105263	51.578947368421105	46.842105263157904	45.789473684210

Table 58 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	41.05263157894474	44.7368421052632	51.052631578947405	48.421052631578895	42.105263157894704	54.736842105263419	41.05263157894474	43.1578947368421	51.052631578947405
AOT	44.7368421052632	47.8947368421053	49.4736842105263	46.842105263157904	52.1052631578947	44.7368421052632	44.7368421052632	46.842105263157904	49.4736842105263
BA	48.148148148148	46.296296296296305	48.148148148148	62.962962962963	62.962962962963	64.814814814814481	42.592592592592595	42.592592592592595	48.148148148148
BANPU	48.9473684210526	48.9473684210526	50.5263157894737	44.2105263157895	45.263157894737	44.7368421052632	49.4736842105263	49.4736842105263	50.5263157894737
BBL	53.1578947368421	50.0	50.0	47.36842105263316	44.736842105263	48.947368421052	52.631578947368396	51.052631578947405	50.0
BCP	46.31578947368421	44.2105263157895	50.0	47.36842105263316	45.263157894737	48.421052631578895	47.8947368421053	44.7368421052632	50.0
BDMS	50.0	49.4736842105263	52.631578947368396	48.421052631578895	48.9473684210526	49.4736842105263	49.4736842105263	47.8947368421053	52.1052631578947
BEC	57.894736842105296	56.842105263157904	51.578947368421105	51.052631578947405	48.421052631578895	51.052631578947405	58.421052631578995	56.31578947368421	51.578947368421105
BH	53.1578947368421	50.5263157894737	54.2105263157895	47.36842105263316	47.8947368421053	48.947368421052	52.1052631578947	51.578947368421105	54.2105263157895
BLA	50.0	49.4736842105263	49.4736842105263	44.7368421052632	49.4736842105263	48.421052631578895	47.8947368421053	48.421052631578895	49.4736842105263
BTS	51.578947368421105	46.31578947368421	51.578947368421105	46.842105263157904	43.684210526315795	51.052631578947405	50.5263157894737	46.31578947368421	51.578947368421105
CBG	40.0	44.0	52.0	57.99999999999999	52.0	54.0	40.0	44.0	54.0
CENTEL	40.0	45.2631578947368	47.8947368421053	49.4736842105263	47.36842105263316	50.0	40.0	46.31578947368421	47.8947368421053
CK	44.7368421052632	48.421052631578895	45.2631578947368	52.1052631578947	52.1052631578947	51.052631578947405	44.7368421052632	48.421052631578895	45.2631578947368
CPALL	46.31578947368421	44.2105263157895	45.7894736842105	43.684210526315795	43.157894736842105	46.842105263157904	44.7368421052632	44.2105263157895	46.31578947368421
CPF	47.8947368421053	52.631578947368396	51.052631578947405	51.578947368421105	50.0	51.052631578947405	48.9473684210526	53.1578947368421	51.052631578947405
CPN	51.578947368421105	52.631578947368396	56.842105263157904	43.684210526315795	43.157894736842105	44.2105263157895	52.1052631578947	50.5263157894737	56.842105263157904
DELTA	45.7894736842105	45.7894736842105	45.7894736842105	48.421052631578895	46.842105263157904	48.947368421052	44.2105263157895	46.842105263157904	45.7894736842105
DTAC	52.1052631578947	49.4736842105263	47.8947368421053	50.5263157894737	54.2105263157895	52.631578947368396	52.631578947368396	50.5263157894737	47.36842105263316
EGCO	43.1578947368421	46.842105263157904	51.578947368421105	38.9473684210526	41.5789473684211	40.5263157894737	46.3157894736842	46.842105263157904	51.052631578947405
GLOW	47.36842105263316	46.842105263157904	46.31578947368421	44.2105263157895	43.684210526315795	45.263157894737	47.8947368421053	44.2105263157895	46.31578947368421
GPSC	37.5	56.25	50.0	43.75	43.75	43.75	43.75	62.5	50.0
HMPRO	54.2105263157895	56.31578947368421	52.631578947368396	46.31578947368421	50.0	48.421052631578895	55.7894736842105	56.31578947368421	52.631578947368396
INTUCH	44.7368421052632	41.57894736842111	47.8947368421053	50.5263157894737	48.421052631578895	52.1052631578947	44.7368421052632	42.631578947368396	47.8947368421053
IRPC	42.105263157894704	45.7894736842105	48.421052631578895	46.842105263157904	44.2105263157895	54.736842105263419	43.1578947368421	44.2105263157895	48.421052631578895
IVL	52.1052631578947	49.4736842105263	57.368421052631604	54.736842105263419	51.578947368421105	52.1052631578947	52.631578947368396	51.052631578947405	57.368421052631604
KBANK	49.4736842105263	47.3684210526316	53.157894736421	46.842105263157904	47.8947368421053	51.578947368421105	49.4736842105263	46.842105263157904	53.157894736842
KCE	53.1578947368421	51.052631578947405	54.736842105263419	43.684210526315795	38.4210526315789	43.157894736842	52.631578947368396	51.052631578947405	54.736842105263419
KTB	41.05263157894474	40.0	37.894736842105296	43.157894736842	42.105263157894704	45.789473684105	40.5263157894737	40.0	37.894736842105296
LH	48.9473684210526	55.2631578947368	59.4736842105263	45.789473684210 5	47.8947368421053	43.157894736842 1	48.9473684210526	54.2105263157895	60.52631578947369
MINT	45.263157894736 8	50.5263157894737	46.3157894736842	51.578947368421105	47.8947368421053	48.947368421052 6	53.1578947368421	50.0	46.31578947368421
MTLS	55.1020408163265	53.0612244897959	51.020408163265294	57.142857142857096	46.9387755102041	48.9795918367347	53.0612244897959	53.0612244897959	51.020408163265294
PS	53.1578947368421	50.0	53.1578947368421	50.0	51.578947368421105	53.1578947368421	50.5263157894737	50.5263157894737	53.684210526315 8
PTT	51.052631578947405	51.052631578947405	54.2105263157895	45.2631578947368	45.789473684210 5	48.421052631578895	53.1578947368421	50.5263157894737	54.736842105263419
PTTEP	53.1578947368421	51.052631578947405	52.631578947368396	51.052631578947405	49.4736842105263	45.263157894736 8	51.578947368421105	51.578947368421105	52.631578947368396
PTTGC	52.631578947368396	51.052631578947405	48.9473684210526	54.2105263157895	49.4736842105263	52.631578947368396	53.1578947368421	50.5263157894737	48.9473684210526
ROBINS	50.0	50.5263157894737	52.631578947368396	54.2105263157895	54.736842105263419	51.578947368421105	50.5263157894737	50.0	52.631578947368396
SAWAD	43.333333333333 3	46.66666666666667	42.22222222222222	56.66666666666667	50.0	53.333333333333 3	43.333333333333 3	45.555555555556	43.333333333333 3
SCB	50.5263157894737	51.052631578947405	49.4736842105263	48.9473684210526	48.9473684210526	48.947368421053	50.5263157894737	51.578947368421105	50.0
SCC	47.36842105263316	47.3684210526316	50.0	48.9473684210526	52.1052631578947	50.5263157894737	48.421052631578895	47.8947368421053	50.0
TASCO	52.631578947368396	50.0	48.421052631578895	45.2631578947368	45.789473684210 5	46.3157894736842	53.1578947368421	49.4736842105263	48.421052631578895
TCAP	54.2105263157895	51.578947368421105	52.631578947368396	48.421052631578895	50.0	52.631578947368396	53.684210526315 8	52.1052631578947	53.157894736421
TMB	53.1578947368421	48.9473684210526	48.9473684210526	44.7368421052632	49.4736842105263	46.842105263157904	51.578947368421105	50.5263157894737	48.9473684210526
TOP	51.578947368421105	46.31578947368421	49.4736842105263	51.578947368421105	49.4736842105263	51.578947368421105	51.052631578947405	45.789473684210 5	49.4736842105263
TPIPL	51.052631578947405	48.9473684210526	51.578947368421105	52.631578947368396	54.2105263157895	50.5263157894737	52.631578947368396	48.9473684210526	51.578947368421105
TRUE	45.789473684210 5	47.3684210526316	48.421052631578895	52.1052631578947	51.052631578947405	50.0	45.263157894736 8	47.3684210526316	48.421052631578895
TTW	47.36842105263316	49.4736842105263	43.684210526315795	44.7368421052632	41.052631578947 4	43.684210526315795	46.842105263157904	48.421052631578895	43.684210526315795
TU	47.8947368421053	43.1578947368421	47.3684210526316	46.31578947368421	46.3157894736842	42.631578947368396	47.3684210526316	42.1052631578947 4	46.842105263157904
WHA	50.0	47.8947368421053	45.789473684210 5	51.578947368421105	48.421052631578895	46.842105263157904	50.0	45.789473684210 5	46.842105263157904

Table 59 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.631578947368396	43.684210526315795	46.842105263157904	42.631578947368396	44.7368421052632	51.578947368421105	42.631578947368396	43.684210526315795	46.842105263157904
AOT	39.473684210526 3	40.526315789473 7	38.947368421052 6	45.789473684210 5	44.736842105263 2	42.105263157894704	39.473684210526 3	40.526315789473 7	38.947368421052 6
BA	40.740740740740 7	38.888888888888 9	40.740740740740 7	68.518518518518 5	48.148148148148 1	46.296296296296305	40.740740740740 7	38.888888888888 9	42.592592592592595
BANPU	37.894736842105296	36.842105263157 9	38.947368421052 6	58.421052631578995	51.052631578947405	53.157894736842 1	37.894736842105296	36.842105263157 9	38.947368421052 6
BBL	48.421052631578895	46.842105263157904	48.421052631578895	44.2105263157895	55.789473684210 5	42.631578947368396	48.421052631578895	46.842105263157904	48.421052631578895
BCP	40.0	40.0	40.0	59.473684210526 3	63.684210526315 8	55.263157894736 8	40.0	40.0	40.0
BDMS	39.473684210526 3	39.473684210526 3	37.894736842105296	59.473684210526 3	56.842105263157904	62.631578947368 4	39.473684210526 3	39.473684210526 3	37.894736842105296
BEC	50.526315789473 7	51.578947368421105	50.526315789473 7	50.0	47.368421052631 6	50.526315789473 7	50.526315789473 7	51.578947368421105	50.526315789473 7
BH	46.842105263157904	45.789473684210 5	46.842105263157904	51.052631578947405	43.684210526315795	43.684210526315795	46.842105263157904	45.789473684210 5	46.842105263157904
BLA	50.526315789473 7	44.210526315789 5	47.368421052631 6	47.368421052631 6	45.263157894736 8	49.473684210526 3	50.526315789473 7	44.210526315789 5	47.368421052631 6
BTS	41.578947368421 1	40.0	40.0	44.2105263157895	39.473684210526 3	52.105263157894 7	41.578947368421 1	40.0	40.0
CBG	42.0	40.0	42.0	44.0	50.0	42.0	38.0	40.0	42.0
CENTEL	37.894736842105296	38.421052631578 9	39.473684210526 3	56.842105263157904	61.052631578947405	54.736842105263 9	37.894736842105296	38.421052631578 9	39.473684210526 3
CK	40.0	38.947368421052 6	38.947368421052 6	42.631578947368396	46.842105263157904	51.052631578947405	40.0	38.947368421052 6	38.947368421052 6
CPALL	41.578947368421 1	41.578947368421 1	43.157894736842 1	52.631578947368396	57.894736842105296	57.368421052631604	41.578947368421 1	41.578947368421 1	43.157894736842 1
CPF	43.157894736842 1	41.578947368421 1	42.631578947368396	43.157894736842 1	43.684210526315795	55.263157894736 8	43.157894736842 1	41.578947368421 1	42.631578947368396
CPN	44.736842105263 2	48.947368421052 6	38.947368421052 6	51.052631578947405	56.842105263157904	56.842105263157904	44.736842105263 2	48.947368421052 6	38.947368421052 6
DELTA	50.0	50.0	50.0	42.631578947368396	46.842105263157904	41.052631578974	50.0	50.0	50.0
DTAC	53.1578947368421	53.1578947368421	52.631578947368396	50.526315789473 7	47.368421052631 6	47.894736842105 3	53.1578947368421	53.1578947368421	52.631578947368396
EGCO	37.368421052631604	37.368421052631604	37.368421052631604	46.315789473684 2	57.368421052631604	54.210526315789 5	37.368421052631604	37.368421052631604	37.368421052631604
GLOW	48.421052631578895	46.842105263157904	46.315789473684 2	51.578947368421105	49.473684210526 3	41.578947368421 1	48.421052631578895	46.842105263157904	46.315789473684 2
GPSC	37.5	43.75	37.5	56.25	56.25	37.5	37.5	43.75	37.5
HMPRO	36.842105263157 9	36.842105263157 9	36.842105263157 9	43.684210526315795	40.526315789473 7	50.0	36.842105263157 9	36.842105263157 9	36.842105263157 9
INTUCH	43.684210526315795	44.210526315789 5	44.736842105263 2	52.631578947368396	40.526315789473 7	45.263157894736 8	43.684210526315795	44.210526315789 5	44.736842105263 2
IRPC	45.789473684210 5	48.421052631578895	43.684210526315795	43.157894736842 1	52.526315789473 7	53.684210526315 8	45.789473684210 5	48.421052631578895	43.684210526315795
IVL	44.736842105263 2	46.315789473684 2	46.315789473684 2	57.368421052631604	52.105263157894 7	56.315789473684 2	44.736842105263 2	46.315789473684 2	46.315789473684 2
KBANK	50.526315789473 7	46.842105263157904	52.105263157894 7	52.631578947368396	52.105263157894 7	56.842105263157904	50.526315789473 7	46.842105263157904	52.105263157894 7
KCE	43.684210526315795	43.157894736842 1	43.684210526315795	56.315789473684 2	53.157894736842 1	57.368421052631604	43.684210526315795	43.157894736842 1	43.684210526315795
KTB	40.526315789473 7	40.526315789473 7	39.473684210526 3	63.157894736842 1	51.052631578947405	52.631578947368396	40.526315789473 7	40.526315789473 7	39.473684210526 3
LH	38.947368421052 6	39.473684210526 3	38.421052631578 9	57.368421052631604	61.578947368421 1	53.684210526315 8	38.947368421052 6	39.473684210526 3	38.421052631578 9
MINT	41.578947368421 1	41.578947368421 1	41.578947368421 1	45.789473684210 5	47.894736842105 3	50.0	41.578947368421 1	41.578947368421 1	41.578947368421 1
MTLS	48.979591836734 7	53.061224489795 9	48.979591836734 7	57.142857142857096	38.775510204081 6	40.816326530612 2	48.979591836734 7	53.061224489795 9	48.979591836734 7
PS	38.947368421052 6	40.0	38.947368421052 6	55.789473684210 5	62.105263157894 7	61.578947368421 1	38.947368421052 6	40.0	38.947368421052 6
PTT	53.1578947368421	47.8947368421053	50.0	52.631578947368396	53.684210526315 8	57.894736842105296	53.1578947368421	47.8947368421053	50.0
PTTEP	50.526315789473 7	53.1578947368421	52.105263157894 7	54.210526315789 5	54.736842105263 9	48.421052631578895	50.526315789473 7	53.1578947368421	52.105263157894 7
PTTGC	51.052631578947405	50.0	50.0	52.105263157894 7	53.684210526315 8	50.0	51.052631578947405	50.0	50.0
ROBINS	42.105263157894704	42.105263157894704	44.736842105263 2	42.631578947368396	37.894736842105296	44.210526315789 5	42.105263157894704	42.105263157894704	44.736842105263 2
SAWAD	37.777777777778	42.222222222222	40.0	50.0	48.888888888889	51.111111111109	37.777777777778	42.222222222222	40.0
SCB	43.1578947368421	41.052631578947 4	41.052631578947 4	56.842105263157904	56.842105263157904	56.315789473684 2	43.1578947368421	41.052631578947 4	41.052631578947 4
SCC	48.421052631578895	45.263157894736 8	48.421052631578895	49.473684210526 3	53.157894736842 1	47.894736842105 3	48.421052631578895	45.263157894736 8	48.421052631578895
TASCO	48.421052631578895	48.421052631578895	48.421052631578895	50.526315789473 7	51.052631578947405	50.526315789473 7	48.421052631578895	48.421052631578895	48.421052631578895
TCAP	38.421052631578 9	38.421052631578 9	38.421052631578 9	51.052631578947405	40.0	54.210526315789 5	38.421052631578 9	38.421052631578 9	38.421052631578 9
TMB	43.684210526315795	44.210526315789 5	43.684210526315795	51.578947368421105	45.789473684210 5	47.368421052631 6	43.684210526315795	44.210526315789 5	43.684210526315795
TOP	53.157894736842 1	49.473684210526 3	52.105263157894 7	57.894736842105296	55.263157894736 8	60.526315789473 69	53.157894736842 1	49.473684210526 3	52.105263157894 7
TPIPL	50.0	48.421052631578895	49.473684210526 3	51.578947368421105	51.578947368421105	56.842105263157904	50.0	48.421052631578895	49.473684210526 3
TRUE	46.315789473684 2	47.368421052631 6	45.263157894736 8	47.368421052631 6	48.421052631578895	46.842105263157904	46.315789473684 2	47.368421052631 6	44.210526315789 5
TTW	34.736842105263 2	34.736842105263 2	34.736842105263 2	42.105263157894704	40.0	52.631578947368396	34.736842105263 2	34.736842105263 2	34.736842105263 2
TU	43.157894736842 1	41.052631578947 4	40.526315789473 7	40.0	43.157894736842 1	40.0	43.157894736842 1	41.052631578947 4	40.526315789473 7
WHA	55.789473684210 5	54.210526315789 5	52.105263157894 7	47.368421052631 6	55.789473684210 5	41.052631578974 5	55.789473684210 5	54.210526315789 5	52.105263157894 7

PredictDirection3 : EMA

Table 60 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	42.631578947368396	44.7368421052632	43.684210526315795	43.684210526315795	44.7368421052632	43.684210526315795	43.1578947368421	44.7368421052632
AOT	60.52631578947369	60.52631578947369	60.52631578947369	60.0	60.52631578947369	60.52631578947369	60.0	60.52631578947369	60.52631578947369
BA	35.185185185185205	37.037037037037	46.296296296296305	44.4444444444444	46.296296296296305	55.5555555555556	33.33333333333333	35.185185185185205	44.4444444444444
BANPU	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474
BBL	45.26315789473368	45.26315789473368	45.7894736842105	44.7368421052632	44.7368421052632	45.26315789473368	45.26315789473368	45.26315789473368	45.7894736842105
BCP	40.52631578947737	45.7894736842105	50.52631578947737	40.52631578947737	43.684210526315795	51.578947368421105	44.7368421052632	42.631578947368396	50.52631578947737
BDMS	60.526315789473369	60.52631578947369	58.9473684210526	60.0	58.9473684210526	58.9473684210526	61.052631578947405	61.5789473684211	58.9473684210526
BEC	45.26315789473368	45.26315789473368	45.7894736842105	45.263157894368	45.263157894368	46.315789473684	44.7368421052632	45.263157894368	45.7894736842105
BH	50.52631578947737	50.52631578947737	54.736842105263159	50.52631578947737	50.52631578947737	48.947368421052632	51.052631578947405	50.52631578947737	50.0
BLA	51.578947368421105	52.1052631578947	48.947368421052632	49.473684210526316	50.52631578947737	48.947368421052632	51.578947368421105	51.052631578947405	48.947368421052632
BTS	46.31578947368421	45.26315789473368	50.52631578947737	46.842105263157904	45.26315789473368	51.578947368421105	46.842105263157904	44.7368421052632	49.473684210526316
CBG	68.0	68.0	70.0	68.0	68.0	70.0	68.0	68.0	70.0
CENTEL	58.421052631578995	58.947368421052632	58.421052631578995	59.473684210526316	58.947368421052632	58.421052631578995	58.421052631578995	58.947368421052632	58.421052631578995
CK	51.052631578947405	58.421052631578995	54.736842105263159	52.1052631578947	57.894736842105296	56.31578947368421	52.1052631578947	58.947368421052632	54.736842105263159
CPALL	55.263157894737368	53.157894736842105	54.2105263157895	54.2105263157895	52.1052631578947	52.1052631578947	54.736842105263159	52.1052631578947	54.2105263157895
CPF	51.578947368421105	51.578947368421105	54.736842105263159	52.1052631578947	51.578947368421105	54.736842105263159	52.1052631578947	52.631578947368396	55.789473684210526
CPN	56.31578947368421	54.736842105263159	55.789473684210526	55.263157894368	57.894736842105296	56.842105263157904	56.31578947368421	56.315789473684	55.789473684210526
DELTA	53.157894736842105	51.052631578947405	50.52631578947737	53.157894736842105	53.157894736842105	49.473684210526316	52.631578947368396	50.52631578947737	50.52631578947737
DTAC	50.0	50.0	50.52631578947737	50.0	50.0	50.52631578947737	50.0	50.0	51.052631578947405
EGCO	58.421052631578995	57.894736842105296	56.842105263157904	57.894736842105296	58.947368421052632	58.947368421052632	56.842105263157904	58.421052631578995	56.842105263157904
GLOW	47.89473684210526	45.263157894368	43.157894736842105	44.7368421052632	48.421052631578895	47.894736842105263	47.368421052632	46.315789473684	44.2105263157895
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	54.736842105263159	55.263157894737368	54.736842105263159	55.263157894737368	54.2105263157895	50.52631578947737	54.736842105263159	52.631578947368396	54.736842105263159
INTUCH	42.631578947368396	41.57894736842211	43.157894736842105	42.631578947368396	41.57894736842211	42.105263157894704	42.105263157894704	41.57894736842211	43.157894736842105
IRPC	55.263157894737368	54.2105263157895	54.2105263157895	55.263157894737368	55.263157894737368	53.157894736842105	55.263157894737368	55.263157894737368	54.2105263157895
IVL	53.684210526315758	54.736842105263159	55.789473684210526	54.2105263157895	53.684210526315758	52.1052631578947	54.736842105263159	54.736842105263159	55.789473684210526
KBANK	43.157894736842105	41.05263157894474	46.842105263157904	44.2105263157895	44.2105263157895	46.315789473684	42.631578947368396	42.105263157894704	46.842105263157904
KCE	60.0	60.0	59.473684210526316	60.0	60.0	60.0	60.0	60.0	59.473684210526316
KTB	41.57894736842211	41.05263157894474	39.473684210526316	41.57894736842211	41.57894736842211	39.473684210526316	41.05263157894474	40.526315789473737	39.473684210526316
LH	41.05263157894474	45.7894736842105	50.0	39.473684210526316	39.473684210526316	50.0	40.0	45.263157894737368	48.947368421052632
MINT	57.894736842105296	57.894736842105296	57.894736842105296	58.421052631578995	58.421052631578995	57.894736842105296	58.421052631578995	57.894736842105296	57.368421052631604
MTLS	51.020408163265294	53.061224897959	55.102040816326516	51.020408163265294	53.061224897959	55.102040816326516	48.979591836734753	53.061224897959	55.102040816326516
PS	37.894736842105296	37.368421052631604	64.210526315795	37.894736842105296	37.894736842105296	64.210526315795	38.421052631578995	37.368421052631604	64.736842105263158
PTT	44.736842105263632	43.684210526315795	44.7368421052632	44.7368421052632	44.2105263157895	44.7368421052632	44.7368421052632	44.2105263157895	44.736842105263632
PTTEP	47.36842105263316	47.36842105263316	46.842105263157904	47.36842105263316	47.36842105263316	46.842105263157904	47.36842105263316	47.36842105263316	46.842105263157904
PTTGC	47.894736842105	48.947368421052632	48.421052631578895	48.947368421052632	47.36842105263316	47.36842105263316	48.421052631578895	48.421052631578895	47.368421052631616
ROBINS	57.894736842105296	54.2105263157895	60.0	56.315789473684232	56.315789473684232	58.947368421052632	56.842105263157904	56.842105263157904	60.0
SAWAD	47.777777777778	47.777777777778	44.444444444444	46.666666666667	47.777777777778	47.777777777778	48.888888888889	47.777777777778	43.333333333333
SCB	40.52631578947737	40.52631578947737	41.05263157894474	40.0	40.52631578947737	41.05263157894474	40.52631578947737	40.52631578947737	41.05263157894474
SCC	42.631578947368396	42.105263157894704	46.315789473684232	41.578947368421	43.157894736842105	46.842105263157904	42.105263157894704	43.157894736842105	45.789473684210526
TASCO	53.157894736842105	53.684210526315758	53.684210526315758	53.684210526315758	53.684210526315758	53.684210526315758	53.684210526315758	53.684210526315758	54.210526315789505
TCAP	54.736842105263159	55.789473684210526	56.842105263157904	55.789473684210526	56.842105263157904	57.368421052631604	54.736842105263159	56.315789473684232	56.842105263157904
TMB	46.842105263157904	45.263157894737368	47.368421052631604	46.315789473684232	46.315789473684232	46.315789473684232	46.842105263157904	48.421052631578895	47.368421052631604
TOP	60.526315789473369	60.526315789473369	59.473684210526316	60.526315789473369	60.0	59.473684210526316	60.526315789473369	60.526315789473369	59.473684210526316
TPIPL	49.473684210526316	48.421052631578895	49.473684210526316	47.894736842105316	49.473684210526316	52.105263157894704	50.0	46.842105263157904	50.0
TRUE	46.842105263157904	46.315789473684232	53.684210526315758	47.368421052631616	47.368421052631616	53.684210526315758	46.842105263157904	47.368421052631616	53.684210526315758
TTW	35.263157894737368	35.263157894737368	55.263157894737368	35.263157894737368	34.736842105263232	58.421052631578995	35.263157894737368	35.789473684210526	55.263157894737368
TU	48.947368421052626	48.421052631578895	46.842105263157904	51.052631578947405	48.421052631578895	47.368421052631616	49.473684210526316	49.473684210526316	46.842105263157904
WHA	45.263157894737368	45.7894736842105	47.368421052631616	45.7894736842105	45.263157894737368	50.0	44.2105263157895	46.315789473684232	46.315789473684232

Table 61 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	54.736842105263159	55.263157894737368	56.842105263157904	43.1578947368421	43.157894736842105	43.684210526315795	43.684210526315795	50.0	57.368421052631604
AOT	42.105263157894704	46.842105263157904	53.684210526315758	44.2105263157895	58.947368421052632	60.0	57.894736842105296	46.842105263157904	53.684210526315758
BA	42.592592592595	37.037037037037	38.888888888889	35.185185185185205	50.0	61.111111111111	46.296296296305	37.037037037037	40.7407407407407
BANPU	41.57894736842211	51.052631578947405	39.473684210526316	60.0	41.05263157894474	45.263157894737368	41.05263157894474	51.578947368421105	39.473684210526316
BBL	44.736842105263632	42.631578947368396	44.7368421052632	59.473684210526316	57.894736842105296	44.736842105263632	44.736842105263632	42.631578947368396	44.736842105263632
BCP	45.263157894737368	44.736842105263632	43.684210526315795	38.421052631578995	36.842105263157904	40.526315789473737	45.789473684210526	44.736842105263632	43.684210526315795
BDMS	61.57894736842211	63.684210526315758	54.736842105263159	50.526315789473737	58.421052631578995	56.315789473684232	61.578947368421105	63.684210526315758	54.210526315789505
BEC	51.578947368421105	54.736842105263159	46.315789473684232	45.789473684210526	47.894736842105263	51.578947368421105	53.157894736842105	54.736842105263159	47.368421052631616
BH	49.473684210526316	47.894736842105263	47.368421052631616	48.421052631578895	45.789473684210526	50.526315789473737	51.052631578947405	47.894736842105263	47.894736842105263
BLA	54.736842105263159	50.526315789473737	49.473684210526316	48.421052631578895	50.0	50.526315789473737	47.894736842105263	50.526315789473737	49.473684210526316
BTS	52.631578947368396	44.210526315789505	44.210526315789505	41.05263157894474	41.578947368421105	48.947368421052632	49.473684210526316	43.157894736842105	44.736842105263632
CBG	34.0	32.0	32.0	34.0	34.0	56.000000000000001	34.0	36.0	32.0
CENTEL	45.263157894737368	58.421052631578995	58.421052631578995	47.894736842105316	58.421052631578995	57.368421052631604	45.263157894737368	58.421052631578995	58.421052631578995
CK	46.315789473684232	46.315789473684232	44.736842105263632	46.315789473684232	48.947368421052632	55.263157894737368	46.842105263157904	45.789473684210526	44.736842105263632
CPALL	50.526315789473737	52.631578947368396	55.263157894737368	38.947368421052632	40.526315789473737	45.789473684210526	50.526315789473737	51.578947368421105	55.263157894737368
CPF	56.315789473684232	57.894736842105296	55.789473684210526	56.315789473684232	56.315789473684232	55.789473684210526	56.315789473684232	58.421052631578995	55.789473684210526
CPN	49.473684210526316	52.631578947368396	56.842105263157904	47.368421052631616	47.894736842105263	44.210526315789505	50.0	52.631578947368396	55.789473684210526
DELTA	41.57894736842211	44.210526315789505	41.05263157894474	43.684210526315795	41.052631578947474	42.105263157894704	42.631578947368396	43.157894736842105	41.052631578947474
DTAC	54.736842105263159	53.684210526315758	51.578947368421105	50.526315789473737	53.157894736842105	51.578947368421105	54.210526315789505	52.631578947368396	50.526315789473737
EGCO	51.052631578947405	54.736842105263159	56.315789473684232	53.684210526315758	52.631578947368396	52.631578947368396	51.052631578947405	54.210526315789505	55.789473684210526
GLOW	45.789473684210526	40.526315789473737	42.631578947368396	41.052631578947474	40.526315789473737	42.631578947368396	42.631578947368396	40.526315789473737	42.631578947368396
GPSC	43.75	37.5	68.75	43.75	43.75	43.75	37.5	37.5	56.25
HMPRO	51.052631578947405	49.473684210526316	47.894736842105316	50.0	42.631578947368396	42.631578947368396	52.1052631578947	48.947368421052632	47.894736842105263
INTUCH	43.157894736842105	42.631578947368396	41.57894736842211	42.631578947368396	43.157894736842105	42.631578947368396	43.157894736842105	42.631578947368396	41.57894736842211
IRPC	47.894736842105263	47.894736842105263	52.631578947368396	47.368421052631616	48.421052631578895	52.631578947368396	48.947368421052632	49.473684210526316	52.631578947368396
IVL	50.526315789473737	50.526315789473737	47.368421052631616	42.631578947368396	44.210526315789505	45.263157894737368	51.578947368421105	50.526315789473737	47.368421052631616
KBANK	49.473684210526263	45.789473684210526	44.210526315789505	45.789473684210526	43.157894736842105	49.473684210526263	47.894736842105263	46.315789473684232	44.210526315789505
KCE	46.315789473684232	46.842105263157904	58.421052631578995	40.0	40.0	40.0	44.736842105263632	42.105263157894704	57.894736842105296
KTB	40.526315789473737	40.526315789473737	44.736842105263632	38.421052631578995	38.947368421052632	42.105263157894704	40.526315789473737	40.526315789473737	44.736842105263632
LH	54.736842105263159	58.947368421052632	58.421052631578995	50.526315789473737	39.473684210526263	47.894736842105263	58.947368421052632	59.473684210526263	61.052631578947405
MINT	49.473684210526263	57.894736842105296	57.368421052631604	52.631578947368396	56.842105263157904	57.894736842105296	49.473684210526263	57.894736842105296	57.368421052631604
MTLS	46.938775510204153	51.020408163265294	46.938775510204153	44.897959183673469	48.979591836734753	51.020408163265294	53.061224897959	51.020408163265294	46.938775510204153
PS	37.894736842105296	36.315789473684232	36.315789473684232	36.842105263157904	34.736842105263232	36.315789473684232	37.894736842105296	37.368421052631604	35.789473684210526
PTT	55.263157894737368	52.1052631578947	51.052631578947405	47.894736842105316	41.052631578947474	48.421052631578895	51.052631578947405	48.421052631578895	51.052631578947405
PTTEP	53.157894736842105	53.157894736842105	51.578947368421105	56.315789473684232	55.789473684210526	54.210526315789505	52.631578947368396	52.631578947368396	50.526315789473737
PTTGC	50.0	44.210526315789505	45.789473684210526	47.368421052631616	51.052631578947405	54.210526315789505	47.368421052631616	44.210526315789505	45.789473684210526
ROBINS	57.894736842105296	61.578947368421105	60.0	45.263157894737368	54.736842105263219	59.473684210526263	59.473684210526263	61.578947368421105	59.473684210526263
SAWAD	54.444444444439	54.444444444439	44.444444444444	57.777777777779	52.222222222222	35.555555555556	57.777777777779	53.333333333333	44.444444444444
SCB	47.894736842105263	50.526315789473737	45.789473684210526	40.526315789473737	40.526315789473737	43.684210526315795	50.0	51.052631578947405	45.263157894737368
SCC	42.631578947368396	45.263157894737368	45.263157894737368	44.736842105263632	43.157894736842105	44.210526315789505	41.052631578947474	45.263157894737368	45.789473684210526
TASCO	46.315789473684232	46.315789473684232	46.315789473684232	46.315789473684232	46.315789473684232	46.315789473684232	46.315789473684232	46.315789473684232	46.315789473684232
TCAP	52.631578947368396	50.0	52.105263157894737	51.052631578947405	49.473684210526263	52.105263157894737	52.631578947368396	50.526315789473737	52.105263157894737
TMB	45.263157894737368	45.263157894737368	44.736842105263632	45.789473684210526	42.105263157894704	44.736842105263632	45.263157894737368	45.789473684210526	45.263157894737368
TOP	42.105263157894704	42.105263157894704	46.842105263157904	45.789473684210526	47.368421052631616	48.421052631578895	45.263157894737368	43.684210526315795	48.947368421052626
TPIPL	44.736842105263632	44.210526315789505	38.947368421052626	48.421052631578895	49.473684210526263	47.894736842105316	44.736842105263632	45.263157894737368	38.947368421052626
TRUE	49.473684210526263	45.789473684210526	45.263157894737368	50.0	44.736842105263632	45.263157894737368	49.473684210526263	46.315789473684232	45.263157894737368
TTW	60.0	46.842105263157904	41.052631578947474	44.736842105263632	36.842105263157904	38.421052631578995	60.0	46.315789473684232	41.052631578947474
TU	40.526315789473737	43.684210526315795	54.736842105263219	46.842105263157904	47.894736842105263	47.368421052631616	40.526315789473737	43.157894736842105	54.736842105263219
WHA	47.894736842105263	47.368421052631616	43.157894736842105	51.052631578947405	47.368421052631616	44.736842105263632	45.789473684210526	46.315789473684232	43.157894736842105

Table 62 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	45.2631578947368	43.1578947368421	44.2105263157895	43.684210526315795	43.1578947368421	43.684210526315795	44.7368421052632
AOT	45.2631578947368	42.631578947368396	46.3157894736842	46.3157894736842	50.5263157894737	59.473684210526	41.0526315789474	43.1578947368421	56.3157894736842
BA	44.4444444444444	42.592592592592595	38.88888888888889	40.7407407407407	42.592592592592595	44.4444444444444	44.4444444444444	38.88888888888889	38.88888888888889
BANPU	41.5789473684211	41.5789473684211	39.4736842105263	41.0526315789474	41.0526315789474	44.2105263157895	41.0526315789474	42.105263157894704	39.4736842105263
BBL	44.2105263157895	43.1578947368421	44.7368421052632	46.3157894736842	46.842105263157904	44.2105263157895	44.2105263157895	43.1578947368421	44.2105263157895
BCP	47.3684210526316	45.2631578947368	43.684210526315795	38.4210526315789	36.84210526315789	40.5263157894737	44.7368421052632	44.7368421052632	43.684210526315795
BDMS	61.5789473684211	63.6842105263158	54.73684210526319	50.5263157894737	57.368421052631604	56.3157894736842	61.5789473684211	63.6842105263158	54.2105263157895
BEC	51.578947368421105	54.73684210526319	47.8947368421053	48.4210526315789	45.26315789473684	51.578947368421105	52.1052631578947	54.73684210526319	48.9473684210526
BH	48.421052631578895	48.421052631578895	46.842105263157904	40.0	41.5789473684211	47.3684210526316	45.2631578947368	47.3684210526316	47.3684210526316
BLA	45.7894736842105	50.5263157894737	49.4736842105263	48.4210526315789	49.4736842105263	50.5263157894737	48.9473684210526	48.4210526315789	49.4736842105263
BTS	45.2631578947368	42.631578947368396	45.2631578947368	43.684210526315795	42.105263157894704	47.8947368421053	47.3684210526316	42.631578947368396	44.7368421052632
CBG	32.0	34.0	32.0	34.0	34.0	56.000000000001	32.0	34.0	32.0
CENTEL	45.2631578947368	55.7894736842105	58.421052631578995	45.7894736842105	58.421052631578995	51.578947368421105	45.2631578947368	56.842105263157904	58.421052631578995
CK	41.0526315789474	43.1578947368421	49.4736842105263	49.4736842105263	49.4736842105263	56.3157894736842	40.5263157894737	43.1578947368421	48.9473684210526
CPALL	51.052631578947405	52.1052631578947	55.2631578947368	38.9473684210526	40.5263157894737	45.7894736842105	51.052631578947405	52.631578947368396	55.2631578947368
CPF	58.421052631578995	56.3157894736842	56.3157894736842	56.842105263157904	55.2631578947368	52.631578947368396	56.842105263157904	56.3157894736842	55.7894736842105
CPN	49.4736842105263	53.1578947368421	56.842105263157904	48.4210526315789	47.8947368421053	44.2105263157895	49.4736842105263	52.631578947368396	55.7894736842105
DELTA	42.631578947368396	44.2105263157895	41.0526315789474	41.5789473684211	40.0	42.105263157894704	43.684210526315795	43.684210526315795	41.0526315789474
DTAC	54.73684210526319	53.684210526315	52.1052631578947	50.5263157894737	51.578947368421105	53.1578947368421	54.2105263157895	53.684210526315	51.052631578947405
EGCO	49.4736842105263	54.2105263157895	56.3157894736842	46.842105263157904	50.0	52.631578947368396	50.5263157894737	54.73684210526319	55.7894736842105
GLOW	41.5789473684211	42.105263157894704	42.631578947368396	40.5263157894737	40.5263157894737	42.631578947368396	44.2105263157895	42.631578947368396	42.631578947368396
GPSC	37.5	37.5	68.75	43.75	43.75	43.75	37.5	37.5	56.25
HMPRO	51.578947368421105	49.4736842105263	47.8947368421053	47.8947368421053	42.105263157894704	42.631578947368396	51.052631578947405	50.0	47.8947368421053
INTUCH	42.105263157894704	42.105263157894704	41.5789473684211	42.631578947368396	43.1578947368421	42.631578947368396	42.631578947368396	42.105263157894704	41.5789473684211
IRPC	48.9473684210526	47.3684210526316	52.631578947368396	47.3684210526316	44.2105263157895	53.1578947368421	48.9473684210526	47.3684210526316	52.631578947368396
IVL	51.052631578947405	50.0	41.5789473684211	44.7368421052632	49.4736842105263	43.684210526315795	51.578947368421105	50.5263157894737	41.0526315789474
KBANK	47.8947368421053	45.7894736842105	43.1578947368421	43.684210526315795	43.1578947368421	41.5789473684211	45.2631578947368	47.3684210526316	43.1578947368421
KCE	44.2105263157895	45.2631578947368	58.9473684210526	42.1052631578947	40.0	40.0	44.2105263157895	51.578947368421105	59.4736842105263
KTB	40.0	39.4736842105263	44.7368421052632	38.4210526315789	37.894736842105296	42.105263157894704	40.5263157894737	38.9473684210526	44.7368421052632
LH	56.842105263157904	60.52631578947369	55.7894736842105	50.5263157894737	49.4736842105263	47.8947368421053	57.894736842105296	60.52631578947369	57.894736842105296
MINT	51.052631578947405	51.052631578947405	58.421052631578995	50.0	53.684210526315	59.473684210263	50.0	50.52631578947368	58.421052631578995
MTLS	44.8979591836735	51.020408163265294	46.9387755102041	46.9387755102041	48.9795918367347	51.020408163265294	48.9795918367347	53.0612244897959	46.9387755102041
PS	39.4736842105263	40.0	33.684210526316	35.7894736842105	35.26315789473684	36.3157894736842	37.894736842105296	40.0	33.157894736842096
PTT	55.7894736842105	52.1052631578947	50.5263157894737	41.0526315789474	41.0526315789474	48.421052631578895	54.73684210526319	49.4736842105263	51.052631578947405
PTTEP	53.1578947368421	53.1578947368421	52.1052631578947	55.2631578947368	55.7894736842105	53.1578947368421	53.684210526316	53.1578947368421	52.631578947368396
PTTGC	49.4736842105263	44.2105263157895	45.7894736842105	47.3684210526316	49.4736842105263	54.2105263157895	49.4736842105263	44.2105263157895	45.7894736842105
ROBINS	58.421052631578995	61.578947368421	60.0	42.105263157894704	55.7894736842105	59.473684210263	56.3157894736842	61.052631578947405	59.473684210263
SAWAD	57.77777777777779	54.44444444444439	44.4444444444444	57.77777777777779	52.22222222222222	35.55555555555556	57.77777777777779	53.33333333333333	44.4444444444444
SCB	52.1052631578947	51.052631578947405	45.7894736842105	40.5263157894737	40.5263157894737	43.684210526315795	50.5263157894737	50.5263157894737	45.26315789473684
SCC	42.631578947368396	43.1578947368421	44.7368421052632	44.2105263157895	43.1578947368421	42.105263157894704	41.5789473684211	42.631578947368396	45.7894736842105
TASCO	46.3157894736842	46.3157894736842	46.3157894736842	46.3157894736842	46.3157894736842	46.3157894736842	46.3157894736842	46.3157894736842	46.3157894736842
TCAP	52.1052631578947	50.5263157894737	52.1052631578947	47.8947368421053	51.052631578947405	52.1052631578947	52.1052631578947	50.5263157894737	52.1052631578947
TMB	41.5789473684211	44.7368421052632	45.2631578947368	44.2105263157895	41.0526315789474	45.2631578947368	40.5263157894737	45.2631578947368	45.2631578947368
TOP	45.2631578947368	43.684210526315795	46.842105263157904	44.7368421052632	46.842105263157904	48.421052631578895	45.2631578947368	42.631578947368396	48.9473684210526
TPIPL	43.684210526315795	44.7368421052632	41.5789473684211	45.7894736842105	45.7894736842105	48.9473684210526	42.631578947368396	44.7368421052632	41.5789473684211
TRUE	47.3684210526316	47.8947368421053	44.2105263157895	46.3157894736842	44.7368421052632	47.8947368421053	46.842105263157904	47.8947368421053	45.7894736842105
TTW	62.1052631578947	39.4736842105263	40.5263157894737	38.9473684210526	38.9473684210526	37.894736842105296	59.473684210263	40.5263157894737	40.5263157894737
TU	41.0526315789474	41.5789473684211	54.73684210526319	47.3684210526316	47.3684210526316	47.3684210526316	41.5789473684211	42.631578947368396	54.73684210526319
WHA	46.3157894736842	45.7894736842105	45.2631578947368	52.1052631578947	46.842105263157904	44.7368421052632	46.842105263157904	46.842105263157904	45.2631578947368

Table 63 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	43.684210526315795	43.1578947368421	44.2105263157895	42.631578947368396	43.684210526315795	43.684210526315795	43.684210526315795
AOT	38.9473684210526	38.9473684210526	38.9473684210526	53.684210526315	61.052631578947405	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526
BA	40.7407407407407	40.7407407407407	38.88888888888889	50.0	50.0	57.4074074074074	40.7407407407407	38.88888888888889	38.88888888888889
BANPU	40.0	38.4210526315789	37.894736842105296	39.4736842105263	42.105263157894704	44.7368421052632	40.0	40.0	37.368421052631604
BBL	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	55.2631578947368	44.7368421052632	44.7368421052632	44.7368421052632
BCP	40.0	40.0	38.4210526315789	44.7368421052632	40.0	49.4736842105263	40.0	40.5263157894737	38.4210526315789
BDMS	45.2631578947368	51.578947368421105	42.105263157894704	55.7894736842105	58.9473684210526	39.473684210263	46.842105263157904	54.2105263157895	43.1578947368421
BEC	49.4736842105263	48.421052631578895	45.2631578947368	54.73684210526319	52.631578947368396	45.2631578947368	49.4736842105263	46.842105263157904	45.2631578947368
BH	49.4736842105263	47.3684210526316	48.421052631578895	50.5263157894737	51.578947368421105	49.473684210263	49.473684210263	50.0	47.3684210526316
BLA	47.3684210526316	46.842105263157904	45.7894736842105	45.7894736842105	49.4736842105263	45.7894736842105	46.842105263157904	47.3684210526316	45.7894736842105
BTS	38.9473684210526	38.9473684210526	40.5263157894737	61.052631578947405	40.5263157894737	51.052631578947405	40.5263157894737	38.9473684210526	40.5263157894737
CBG	32.0	32.0	32.0	34.0	32.0	32.0	32.0	32.0	32.0
CENTEL	43.1578947368421	43.1578947368421	41.5789473684211	41.5789473684211	58.421052631578995	51.052631578947405	42.105263157894704	42.631578947368396	41.5789473684211
CK	38.9473684210526	38.9473684210526	38.9473684210526	43.684210526315795	41.5789473684211	43.684210526315795	38.9473684210526	38.9473684210526	38.9473684210526
CPALL	43.1578947368421	43.1578947368421	43.1578947368421	44.2105263157895	53.1578947368421	48.421052631578895	43.1578947368421	43.1578947368421	43.1578947368421
CPF	45.7894736842105	45.7894736842105	43.684210526315795	43.684210526315795	44.2105263157895	46.3157894736842	46.3157894736842	45.2631578947368	43.684210526315795
CPN	42.631578947368396	43.684210526315795	46.3157894736842	46.842105263157904	55.7894736842105	52.1052631578947	42.105263157894704	42.105263157894704	46.3157894736842
DELTA	42.105263157894704	42.105263157894704	45.2631578947368	44.7368421052632	45.2631578947368	52.1052631578947	43.1578947368421	45.2631578947368	45.2631578947368
DTAC	50.0	50.0	48.421052631578895	50.0	50.5263157894737	48.421052631578895	50.5263157894737	50.0	48.421052631578895
EGCO	40.0	41.5789473684211	41.0526315789474	44.2105263157895	51.578947368421105	47.8947368421053	40.5263157894737	40.5263157894737	41.5789473684211
GLOW	41.5789473684211	39.4736842105263	42.105263157894704	46.3157894736842	38.4210526315789	51.052631578947405	42.631578947368396	39.4736842105263	42.105263157894704
GPSC	43.75	43.75	43.75	50.0	43.75	43.75	43.75	43.75	43.75
HMPRO	36.8421052631579	36.8421052631579	36.8421052631579	36.8421052631579	46.842105263157904	41.5789473684211	36.8421052631579	36.8421052631579	36.8421052631579
INTUCH	42.105263157894704	42.105263157894704	42.105263157894704	42.631578947368396	42.105263157894704	43.684210526315795	42.105263157894704	42.105263157894704	42.105263157894704
IRPC	47.3684210526316	45.2631578947368	43.1578947368421	47.3684210526316	52.1052631578947	49.4736842105263	51.052631578947405	45.7894736842105	43.1578947368421
IVL	46.842105263157904	48.421052631578895	47.3684210526316	50.5263157894737	47.3684210526316	56.842105263157904	47.3684210526316	49.4736842105263	47.3684210526316
KBANK	45.2631578947368	45.2631578947368	45.2631578947368	44.2105263157895	46.842105263157904	39.4736842105263	45.2631578947368	45.2631578947368	45.2631578947368
KCE	42.631578947368396	39.4736842105263	40.0	58.421052631578995	43.1578947368421	40.5263157894737	42.105263157894704	41.5789473684211	40.0
KTB	39.4736842105263	39.4736842105263	39.4736842105263	37.894736842105296	39.4736842105263	37.894736842105296	39.4736842105263	40.0	39.4736842105263
LH	42.631578947368396	41.0526315789474	38.4210526315789	38.4210526315789	54.73684210526319	51.578947368421105	43.684210526315795	41.0526315789474	38.4210526315789
MINT	42.105263157894704	42.105263157894704	42.105263157894704	54.2105263157895	52.631578947368396	45.2631578947368	42.105263157894704	42.105263157894704	42.105263157894704
MTLS	46.9387755102041	46.9387755102041	48.9795918367347	51.020408163265294	48.9795918367347	48.9795918367347	51.020408163265294	48.9795918367347	48.9795918367347
PS	36.3157894736842	36.3157894736842	36.3157894736842	37.894736842105296	36.8421052631579	36.3157894736842	36.3157894736842	36.3157894736842	36.3157894736842
PTT	41.0526315789474	41.0526315789474	44.7368421052632	59.473684210263	41.0526315789474	47.8947368421053	41.0526315789474	41.0526315789474	45.2631578947368
PTTEP	52.631578947368396	52.631578947368396	52.631578947368396	51.578947368421105	52.631578947368396	53.1578947368421	54.2105263157895	52.631578947368396	52.631578947368396
PTTGC	46.842105263157904	46.842105263157904	46.842105263157904	57.368421052631604	47.8947368421053	48.9473684210526	46.842105263157904	46.3157894736842	46.842105263157904
ROBINS	37.894736842105296	37.894736842105296	37.894736842105296	50.0	62.1052631578947	38.4210526315789	37.894736842105296	37.894736842105296	37.894736842105296
SAWAD	41.111111111111	41.111111111111	41.111111111111	53.3333333333333	42.2222222222222	38.88888888888889	41.111111111111	42.2222222222222	41.111111111111
SCB	56.3157894736842	43.1578947368421	49.4736842105263	41.0526315789474	41.0526315789474	41.0526315789474	43.684210526315795	42.105263157894704	49.4736842105263
SCC	43.1578947368421	42.105263157894704	40.0	50.5263157894737	43.1578947368421	54.2105263157895	43.684210526315795	42.105263157894704	40.0
TASCO	46.842105263157904	46.3157894736842	46.3157894736842	46.842105263157904	46.3157894736842	54.2105263157895	46.842105263157904	46.3157894736842	46.3157894736842
TCAP	42.105263157894704	39.4736842105263	38.4210526315789	51.578947368421105	55.7894736842105	52.1052631578947	41.0526315789474	41.0526315789474	38.4210526315789
TMB	43.684210526315795	43.684210526315795	43.684210526315795	52.631578947368396	46.842105263157904	48.421052631578895	43.684210526315795	43.684210526315795	43.684210526315795
TOP	45.2631578947368	45.2631578947368	45.2631578947368	42.631578947368396	50.5263157894737	40.0	44.7368421052632	44.7368421052632	45.7894736842105
TPIPL	44.2105263157895	42.105263157894704	41.5789473684211	44.2105263157895	43.684210526315795	41.5789473684211	44.7368421052632	41.5789473684211	41.5789473684211
TRUE	41.5789473684211	43.1578947368421	45.2631578947368	59.473684210263	43.684210526315795	54.736842105263	44.2105263157895	43.1578947368421	45.2631578947368
TTW	34.7368421052632	34.7368421052632	34.7368421052632	37.368421052631604	33.684210526316	47.8947368421053	34.7368421052632	34.7368421052632	34.7368421052632
TU	39.4736842105263	38.9473684210526	40.0	42.631578947368396	45.2631578947368	44.7368421052632	39.4736842105263	40.0	40.0
WHA	44.7368421052632	44.7368421052632	44.7368421052632	46.3157894736842	48.9473684210526	50.0	44.7368421052632	44.7368421052632	44.7368421052632

PredictDirection4 : Volume

Table 64 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	42.631578947368396	43.684210526315795	43.1578947368421	43.684210526315795	44.21052631578947	43.684210526315795	42.631578947368396	43.684210526315795
AOT	58.421052631578995	58.9473684210526	60.52631578947369	58.9473684210526	59.4736842105263	60.52631578947369	59.4736842105263	58.9473684210526	60.52631578947369
BA	48.1481481481481	51.851851851851805	51.851851851851805	48.1481481481481	53.703703703703695	55.55555555555556	46.296296296296305	44.44444444444444	50.0
BANPU	40.0	40.0	43.684210526315795	41.05263157894474	40.0	41.05263157894474	40.5263157894737	40.5263157894737	43.15789473684211
BBL	44.21052631578947	42.631578947368396	42.105263157894704	44.7368421052632	43.15789473684211	44.210526315789 5	43.15789473684211	43.15789473684211	42.631578947368396
BCP	49.4736842105263	52.631578947368396	53.68421052631 58	47.8947368421053	48.421052631578895	50.0	50.5263157894737	52.1052631578947	54.21052631578947
BDMS	60.526315789473 69	59.4736842105263	57.368421052631604	57.368421052631604	59.4736842105263	56.842105263157904	58.9473684210526	59.4736842105263	57.368421052631604
BEC	46.31578947368421	46.31578947368421	45.7894736842105	45.7894736842105	46.31578947368421	45.7894736842105	46.31578947368421	46.31578947368421	45.7894736842105
BH	54.736842105263 19	53.68421052631 58	47.3684210526316	56.31578947368421	52.631578947368396	47.89473684210 53	55.7894736842105	55.7894736842105	47.8947368421053
BLA	48.9473684210526	49.4736842105263	48.9473684210526	48.9473684210526	50.526315789473 7	48.421052631578895	48.9473684210526	49.4736842105263	48.947368421052 6
BTS	54.736842105263 19	52.1052631578947	54.736842105263 19	52.1052631578947	54.21052631578 95	54.21052631578 95	56.31578947368421	52.1052631578947	54.736842105263 19
CBG	70.0	68.0	66.0	68.0	68.0	68.0	68.0	68.0	66.0
CENTEL	57.894736842105296	54.21052631578 95	56.842105263157904	54.736842105263 19	53.684210526315 8	54.736842105263 19	57.368421052631604	54.736842105263 19	56.842105263157904
CK	54.21052631578 95	55.78947368421 05	54.21052631578 95	56.842105263157904	55.263157894736 8	54.21052631578 95	54.736842105263 19	56.31578947368421	54.21052631578 95
CPALL	55.263157894736 8	55.263157894736 8	54.736842105263 19	56.31578947368421	55.789473684210 5	52.631578947368396	55.789473684210 5	55.263157894736 8	54.736842105263 19
CPF	50.526315789473 7	47.89473684210 53	51.578947368421105	52.1052631578947	47.894736842105 3	50.0	51.052631578947405	47.36842105263 16	51.578947368421105
CPN	60.0	57.894736842105296	58.421052631578995	55.78947368421 05	56.31578947368421	56.842105263157904	60.0	58.421052631578995	58.421052631578995
DELTA	53.1578947368421	54.736842105263 19	54.736842105263 19	53.1578947368421	53.684210526315 8	54.736842105263 19	56.31578947368421	54.736842105263 19	54.736842105263 19
DTAC	49.4736842105263	49.4736842105263	49.4736842105263	48.9473684210526	49.4736842105263	49.4736842105263	49.4736842105263	49.4736842105263	49.4736842105263
EGCO	56.31578947368421	55.78947368421 05	56.31578947368421	58.421052631578995	55.263157894736 8	56.31578947368421	57.894736842105296	56.842105263157904	55.78947368421 05
GLOW	55.263157894736 8	53.15789473684 21	53.15789473684 21	50.526315789473 7	51.578947368421105	56.842105263157904	52.1052631578947	52.1052631578947	53.15789473684 21
GPSC	50.0	50.0	43.75	50.0	43.75	43.75	43.75	43.75	43.75
HMPRO	54.736842105263 19	55.263157894736 8	57.894736842105296	55.78947368421 05	55.263157894736 8	57.368421052631604	54.736842105263 19	55.263157894736 8	57.894736842105296
INTUCH	42.105263157894704	44.736842105263 2	41.05263157894 74	42.631578947368396	44.210526315789 5	42.105263157894704	42.631578947368396	44.736842105263 2	41.05263157894 74
IRPC	55.78947368421 05	55.78947368421 05	54.21052631578 95	56.31578947368421	56.842105263157904	55.263157894736 8	55.78947368421 05	56.31578947368421	54.21052631578 95
IVL	56.31578947368421	55.263157894736 8	54.736842105263 19	55.78947368421 05	56.31578947368421	54.736842105263 19	56.842105263157904	55.263157894736 8	54.736842105263 19
KBANK	43.15789473684 21	44.736842105263 2	45.26315789473 68	44.736842105263 2	45.78947368421 05	45.26315789473 68	43.15789473684 21	45.26315789473 68	45.26315789473 68
KCE	59.4736842105263	60.0	60.0	59.4736842105263	60.0	60.0	59.4736842105263	60.0	60.0
KTB	54.21052631578 95	48.947368421052 6	51.578947368421105	52.1052631578947	49.4736842105263	47.89473684210 53	52.1052631578947	50.0	51.578947368421105
LH	48.9473684210526	47.89473684210 53	55.78947368421 05	50.0	48.421052631578895	54.21052631578 95	48.9473684210526	48.9473684210526	55.78947368421 05
MINT	60.0	58.9473684210526	58.421052631578995	58.9473684210526	58.9473684210526	58.421052631578995	59.4736842105263	59.4736842105263	58.421052631578995
MTLS	53.0612244897959	57.142857142857096	53.0612244897959	55.1020408163265	55.1020408163265	57.142857142857096	53.0612244897959	57.142857142857096	55.1020408163265
PS	55.78947368421 05	48.421052631578895	63.68421052631 58	56.842105263157904	48.9473684210526	63.684210526315 8	54.736842105263 19	47.89473684210 53	63.15789473684 21
PTT	44.21052631578 95	42.631578947368396	47.8947368421053	42.105263157894704	42.631578947368396	45.26315789473 68	46.31578947368421	42.631578947368396	47.89473684210 53
PTTEP	51.052631578947405	47.36842105263 16	43.15789473684 21	51.052631578947405	48.421052631578895	43.684210526315795	49.4736842105263	47.36842105263 16	43.15789473684 21
PTTGC	54.21052631578 95	53.68421052631 58	54.21052631578 95	49.4736842105263	53.15789473684 21	55.78947368421 05	53.15789473684 21	53.68421052631 58	54.21052631578 95
ROBINS	49.4736842105263	51.578947368421105	51.578947368421105	51.052631578947405	51.052631578947405	49.4736842105263	50.0	51.578947368421105	52.1052631578947
SAWAD	50.0	51.111111111111109	50.0	51.111111111111109	47.777777777778	50.0	47.777777777778	51.111111111111109	50.0
SCB	43.684210526315795	44.736842105263 2	44.736842105263 2	42.631578947368396	43.15789473684 21	45.78947368421 05	44.210526315789 5	44.736842105263 2	44.736842105263 2
SCC	47.36842105263 16	48.421052631578895	49.4736842105263	46.842105263157904	47.36842105263 16	47.89473684210 53	46.842105263157904	48.421052631578895	50.0
TASCO	50.0	53.15789473684 21	47.36842105263 16	52.1052631578947	55.263157894736 8	47.36842105263 16	48.421052631578895	55.78947368421 05	46.842105263157904
TCAP	53.684210526315 8	52.1052631578947	54.736842105263 19	55.78947368421 05	54.736842105263 19	55.263157894736 8	55.263157894736 8	52.1052631578947	54.736842105263 19
TMB	57.368421052631604	53.68421052631 58	55.263157894736 8	53.68421052631 58	54.736842105263 19	56.842105263157904	55.78947368421 05	54.21052631578 95	55.263157894736 8
TOP	60.0	59.4736842105263	60.0	60.526315789473 69	60.0	59.4736842105263	60.0	59.4736842105263	60.0
TPIPL	55.78947368421 05	49.4736842105263	54.21052631578 95	53.68421052631 58	49.4736842105263	61.052631578947405	55.78947368421 05	49.4736842105263	54.736842105263 19
TRUE	51.578947368421105	47.89473684210 53	54.736842105263 19	54.736842105263 19	45.78947368421 05	53.68421052631 58	54.736842105263 19	48.421052631578895	53.684210526315 8
TTW	53.684210526315 8	54.736842105263 19	61.052631578947405	52.1052631578947	53.15789473684 21	58.421052631578995	54.736842105263 19	52.1052631578947	61.57894736842 11
TU	52.1052631578947	50.526315789473 7	52.1052631578947	50.0	48.947368421052 6	54.736842105263 19	50.526315789473 7	50.526315789473 7	52.1052631578947
WHA	52.1052631578947	55.78947368421 05	57.368421052631604	52.1052631578947	53.68421052631 58	60.0	53.15789473684 21	55.78947368421 05	57.368421052631604

Table 65 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	50.0	48.9473684210526	45.7894736842105	52.1052631578947	51.052631578947405	49.4736842105263	50.0	45.7894736842105	46.31578947368421
AOT	47.36842105263 16	51.052631578947405	51.578947368421105	47.36842105263 16	52.631578947368396	42.631578947368396	50.5263157894737	50.0	51.578947368421105
BA	33.33333333333333	27.777777777778	38.88888888888889	53.703703703703695	50.0	44.44444444444444	33.33333333333333	29.6296296296296	38.88888888888889
BANPU	47.36842105263 16	47.89473684210 53	54.736842105263 19	45.26315789473 68	42.105263157894704	47.36842105263 16	48.421052631578895	47.36842105263 16	54.736842105263 19
BBL	47.89473684210 53	48.9473684210526	47.36842105263 16	48.421052631578895	52.631578947368396	48.421052631578895	47.89473684210 53	49.4736842105263	47.36842105263 16
BCP	52.631578947368396	50.526315789473 7	49.4736842105263	48.421052631578895	51.578947368421105	44.210526315789 5	52.1052631578947	51.052631578947405	49.4736842105263
BDMS	49.4736842105263	47.36842105263 16	42.105263157894704	45.26315789473 68	50.0	38.42105263157 89	47.36842105263 16	46.842105263157904	42.105263157894704
BEC	47.89473684210 53	44.736842105263 2	54.21052631578 95	51.578947368421105	53.15789473684 21	46.842105263157904	46.842105263157904	44.736842105263 2	54.21052631578 95
BH	44.736842105263 2	44.736842105263 2	46.842105263157904	50.526315789473 7	52.631578947368396	50.526315789473 7	44.736842105263 2	42.631578947368396	46.842105263157904
BLA	52.631578947368396	52.1052631578947	48.421052631578895	52.1052631578947	53.68421052631 58	57.894736842105296	52.631578947368396	50.526315789473 7	47.89473684210 53
BTS	55.78947368421 05	53.15789473684 21	50.0	44.736842105263 2	43.15789473684 21	43.684210526315795	54.736842105263 19	52.631578947368396	50.526315789473 7
CBG	46.0	42.0	38.0	52.0	50.0	60.0	48.0	40.0	38.0
CENTEL	44.736842105263 2	49.4736842105263	37.894736842105296	43.684210526315795	42.631578947368396	42.105263157894704	45.78947368421 05	48.9473684210526	37.894736842105296
CK	47.36842105263 16	50.526315789473 7	47.36842105263 16	47.8947368421053	46.31578947368421	41.578947368421 1	48.9473684210526	48.421052631578895	47.36842105263 16
CPALL	51.052631578947405	52.631578947368396	45.26315789473 68	46.842105263157904	44.210526315789 5	46.31578947368421	48.421052631578895	51.578947368421105	47.89473684210 53
CPF	54.21052631578 95	56.842105263157904	56.31578947368421	51.578947368421105	51.052631578947405	47.8947368421053	53.15789473684 21	57.368421052631604	56.31578947368421
CPN	53.15789473684 21	53.68421052631 58	63.15789473684 21	42.631578947368396	46.31578947368421	40.0	54.736842105263 19	53.68421052631 58	63.15789473684 21
DELTA	46.31578947368421	45.26315789473 68	45.78947368421 05	44.7368421052632	44.7368421052632	44.210526315789 5	45.78947368421 05	45.26315789473 68	45.78947368421 05
DTAC	54.21052631578 95	48.421052631578895	53.15789473684 21	50.0	50.0	52.631578947368396	51.578947368421105	47.89473684210 53	54.736842105263 19
EGCO	47.89473684210 53	48.9473684210526	46.31578947368421	49.4736842105263	50.526315789473 7	50.526315789473 7	47.89473684210 53	50.0	45.78947368421 05
GLOW	47.36842105263 16	49.4736842105263	45.26315789473 68	44.7368421052632	47.89473684210 53	43.684210526315795	47.36842105263 16	49.4736842105263	45.78947368421 05
GPSC	50.0	56.25	43.75	50.0	43.75	43.75	56.25	56.25	43.75
HMPRO	54.21052631578 95	50.0	50.0	50.0	47.89473684210 53	42.631578947368396	55.263157894736 8	50.0	50.0
INTUCH	49.4736842105263	47.36842105263 16	43.684210526315795	48.9473684210526	47.89473684210 53	50.526315789473 7	50.526315789473 7	47.36842105263 16	44.210526315789 5
IRPC	56.842105263157904	54.736842105263 19	48.421052631578895	53.684210526315 8	54.736842105263 19	41.578947368421 1	56.842105263157904	56.31578947368421	48.421052631578895
IVL	55.263157894736 8	52.1052631578947	45.78947368421 05	54.21052631578 95	49.4736842105263	53.684210526315 8	52.631578947368396	52.631578947368396	45.78947368421 05
KBANK	50.526315789473 7	47.89473684210 53	47.89473684210 53	48.421052631578895	53.684210526315 8	48.947368421052 6	51.578947368421105	47.89473684210 53	47.36842105263 16
KCE	51.052631578947405	52.631578947368396	50.526315789473 7	43.15789473684 21	45.78947368421 05	40.0	51.578947368421105	53.15789473684 21	50.526315789473 7
KTB	46.31578947368421	51.052631578947405	52.1052631578947	45.78947368421 05	42.631578947368396	45.26315789473 68	46.31578947368421	51.052631578947405	51.578947368421105
LH	46.31578947368421	44.21052631578 95	46.31578947368421	46.31578947368421	43.15789473684 21	45.26315789473 68	45.26315789473 68	44.21052631578 95	46.31578947368421
MINT	44.736842105263 2	49.4736842105263	46.842105263157904	46.31578947368421	45.78947368421 05	41.210526315789 5	46.842105263157904	51.052631578947405	47.36842105263 16
MTLS	42.8571428571429	44.89795918367 35	46.9387755102041	61.2244897959184	63.265306122449005	55.102040816326 5	46.9387755102041	46.9387755102041	42.8571428571429
PS	45.78947368421 05	48.947368421052 6	37.368421052631604	46.842105263157904	53.15789473684 21	46.31578947368421	43.15789473684 21	47.89473684210 53	37.368421052631604
PTT	45.78947368421 05	46.31578947368421	47.36842105263 16	47.36842105263 16	46.31578947368421	41.578947368421 1	45.26315789473 68	46.31578947368421	48.421052631578895
PTTEP	57.368421052631604	51.578947368421105	54.21052631578 95	52.631578947368396	50.0	40.0	57.368421052631604	51.578947368421105	54.21052631578 95
PTTGC	52.1052631578947	48.947368421052 6	52.631578947368396	54.736842105263 19	55.263157894736 8	56.842105263157904	52.631578947368396	48.947368421052 6	52.631578947368396
ROBINS	55.78947368421 05	53.15789473684 21	56.84210526315 7904	52.631578947368396	56.842105263157904	47.36842105263 16	53.68421052631 58	52.631578947368396	56.842105263157904
SAWAD	44.444444444444	47.777777777778	60.0	46.666666666667	53.333333333333	41.111111111111	48.888888888889	46.666666666667	60.0
SCB	50.526315789473 7	51.578947368421105	57.368421052631604	48.947368421052 6	45.78947368421 05	54.210526315789 5	50.526315789473 7	55.263157894736 8	57.894736842105296
SCC	46.31578947368421	43.684210526315795	53.68421052631 58	54.736842105263 19	53.68421052631 58	45.263157894736 8	47.36842105263 16	44.736842105263 2	52.631578947368396
TASCO	44.210526315789 5	44.736842105263 2	44.210526315789 5	45.263157894736 8	44.736842105263 2	46.31578947368421	43.684210526315795	45.263157894736 8	44.210526315789 5
TCAP	49.4736842105263	47.36842105263 16	46.31578947368421	47.36842105263 16	48.421052631578895	49.4736842105263	49.4736842105263	46.842105263157904	46.31578947368421
TMB	51.052631578947405	53.68421052631 58	46.31578947368421	48.947368421052 6	54.210526315789 5	51.578947368421105	52.1052631578947	52.1052631578947	45.78947368421 05
TOP	43.15789473684 21	45.26315789473 68	47.89473684210 53	45.26315789473 68	48.421052631578895	51.578947368421105	44.210526315789 5	46.842105263157904	47.89473684210 53
TPIPL	47.36842105263 16	55.263157894736 8	43.684210526315795	52.631578947368396	56.31578947368421	48.947368421052 6	44.210526315789 5	53.68421052631 58	42.631578947368396
TRUE	48.421052631578895	50.0	46.842105263157904	47.89473684210 53	50.0	46.842105263157904	48.421052631578895	51.052631578947405	46.842105263157904
TTW	43.15789473684 21	51.052631578947405	40.52631578947 37	44.7368421052632	45.26315789473 68	40.52631578947 37	43.684210526315795	51.052631578947405	40.52631578947 37
TU	48.947368421052 6	49.4736842105263	44.210526315795	44.7368421052632	45.26315789473 68	39.473684210526 3	48.947368421052 6	50.0	44.736842105263 2
WHA	48.9473684210526	48.947368421052 6	47.36842105263 16	45.78947368421 05	47.36842105263 16	44.736842105263 2	47.89473684210 53	51.052631578947405	46.842105263157904

Table 66 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	52.631578947368396	48.9473684210526	46.842105263157904	54.73684210526319	57.368421052631604	51.5789473684211105	50.5263157894737	49.4736842105263	47.36842105263316
AOT	53.68421052631158	48.421052631578895	51.052631578947405	41.57894736842111	41.052631578947	38.42105263157895	51.052631578947405	47.89473684210553	51.052631578947405
BA	37.037037037037	40.7407407407407	38.88888888888889	55.55555555555556	50.0	38.88888888888889	33.33333333333333	40.7407407407407	38.88888888888889
BANPU	49.4736842105263	48.9473684210526	55.263157894737368	42.1052631578947	41.052631578947	48.421052631578895	48.421052631578895	48.421052631578895	54.73684210526319
BBL	51.052631578947405	54.2105263157895	46.842105263157904	49.4736842105263	45.263157894737368	44.736842105263 2	50.5263157894737	53.1578947368421	47.36842105263316
BCP	53.1578947368421	50.0	49.4736842105263	48.421052631578895	51.052631578947405	44.2105263157895	53.1578947368421	51.052631578947405	49.4736842105263
BDMS	45.78947368421 05	48.421052631578895	42.1052631578947 04	45.263157894737368	50.5263157894737	38.4210526315789	47.36842105263316	46.842105263157904	42.1052631578947 04
BEC	48.421052631578895	46.3157894736842	55.263157894737368	56.3157894736842	55.263157894737368	48.9473684210526	48.421052631578895	45.78947368421 05	55.263157894737368
BH	47.89473684210553	46.3157894736842	47.36842105263316	50.0	54.2105263157895	49.4736842105263	46.842105263157904	46.3157894736842	47.89473684210553
BLA	53.1578947368421	50.5263157894737	48.421052631578895	52.631578947368396	54.2105263157895	57.894736842105296	52.631578947368396	50.5263157894737	47.8947368421053
BTS	57.368421052631604	51.5789473684211105	43.684210526315795	44.73684210526232	42.1052631578947 04	40.0	56.3157894736842	51.5789473684211105	43.684210526315795
CBG	44.0	42.0	38.0	52.0	50.0	60.0	42.0	38.0	38.0
CENTEL	46.842105263157904	50.5263157894737	40.0	43.684210526315795	43.1578947368421	40.5263157894737	44.2105263157895	50.0	40.0
CK	49.4736842105263	50.0	48.421052631578895	40.5263157894737	45.263157894737368	49.4736842105263	50.0	49.4736842105263	47.36842105263316
CPALL	50.0	52.1052631578947	45.263157894737368	47.36842105263316	44.2105263157895	46.3157894736842	48.9473684210526	52.631578947368396	45.78947368421 05
CPF	53.1578947368421	46.842105263157904	58.94736842105263	55.263157894737368	48.421052631578895	50.0	50.0	50.5263157894737	58.94736842105263
CPN	55.263157894737368	53.1578947368421	63.157894736842	42.631578947368396	45.78947368421 05	40.0	55.78947368421 05	53.684210526315 8	63.157894736842
DELTA	48.421052631578895	45.78947368421 05	45.78947368421 05	45.78947368421 05	45.263157894737368	44.2105263157895	48.421052631578895	45.263157894737368	45.78947368421 05
DTAC	49.4736842105263	53.1578947368421	50.0	50.5263157894737	50.0	55.263157894737368	48.94736842105263	51.578947368421105	48.94736842105263
EGCO	48.94736842105263	48.421052631578895	46.842105263157904	56.3157894736842	51.578947368421105	51.052631578947405	47.89473684210553	48.421052631578895	46.3157894736842
GLOW	46.3157894736842	48.421052631578895	45.78947368421 05	45.78947368421 05	47.3684210526316	44.2105263157895	47.89473684210553	48.421052631578895	46.842105263157904
GPSC	56.25	43.75	43.75	50.0	43.75	43.75	56.25	50.0	43.75
HMPRO	51.578947368421105	52.1052631578947	50.0	46.3157894736842	51.052631578947405	43.1578947368421	55.263157894737368	52.1052631578947	50.0
INTUCH	51.052631578947405	47.36842105263316	43.684210526315795	48.94736842105263	49.4736842105263	50.5263157894737	52.1052631578947	47.36842105263316	44.2105263157895
IRPC	53.1578947368421	54.2105263157895	48.421052631578895	55.263157894737368	53.684210526315 8	41.0526315789474	53.684210526315 8	53.684210526315 8	48.421052631578895
IVL	54.2105263157895	45.263157894737368	50.5263157894737	52.631578947368396	50.0	47.36842105263316	55.263157894737368	45.263157894737368	50.5263157894737
KBANK	50.0	49.4736842105263	45.78947368421 05	48.421052631578895	45.78947368421 05	46.842105263157904	48.421052631578895	45.78947368421 05	44.736842105263 2
KCE	48.421052631578895	52.1052631578947	48.421052631578895	44.2105263157895	45.78947368421 05	40.5263157894737	50.0	51.578947368421 05	48.94736842105263
KTB	46.3157894736842	51.052631578947405	51.578947368421105	44.736842105263 2	44.736842105263 2	45.263157894737368	46.3157894736842	51.578947368421 05	51.578947368421105
LH	47.36842105263316	43.1578947368421	48.94736842105263	47.36842105263316	49.4736842105263	47.89473684210553	47.89473684210553	42.631578947368396	48.94736842105263
MINT	47.89473684210553	45.263157894737368	48.421052631578895	43.1578947368421	42.631578947368396	46.842105263157904	48.421052631578895	45.263157894737368	47.89473684210553
MTLS	44.8979591836735	46.93877551020441	46.93877551020441	65.30612244898796	59.1836734693878	55.1020408163265	42.8571428571429	46.93877551020441	42.8571428571429
PS	49.4736842105263	47.89473684210553	41.578947368421	41.052631578947 4	45.263157894737368	38.4210526315789	48.94736842105263	48.94736842105263	41.052631578947 4
PTT	47.89473684210553	46.3157894736842	47.36842105263316	42.631578947368396	47.36842105263316	41.578947368421 1	48.421052631578895	45.78947368421 05	48.421052631578895
PTTEP	55.263157894737368	52.631578947368396	54.736842105263 19	45.78947368421 05	48.947368421052 6	46.3157894736842	55.263157894737368	53.157894736842 1	54.210526315789 5
PTTGC	52.1052631578947	49.4736842105263	52.631578947368396	55.78947368421 05	54.210526315789 5	56.842105263157904	55.263157894737368	48.9473684210526	52.631578947368396
ROBINS	55.78947368421 05	52.631578947368396	56.842105263157904	52.631578947368396	54.736842105263 19	47.36842105263316	53.684210526315 8	53.684210526315 8	56.842105263157904
SAWAD	50.0	47.77777777777778	60.0	46.66666666666667	47.77777777777778	41.111111111111 1	44.444444444444 4	47.77777777777778	60.0
SCB	50.5263157894737	55.263157894737368	57.368421052631604	48.9473684210526	48.421052631578895	54.210526315789 5	51.052631578947405	55.263157894737368	57.894736842105296
SCC	46.842105263157904	44.736842105263 2	53.6842105263158	43.684210526315795	46.3157894736842	44.2105263157895	47.36842105263316	44.736842105263 2	54.2105263157895
TASCO	45.78947368421 05	43.684210526315795	42.631578947368396	45.263157894737368	46.3157894736842	46.3157894736842	44.736842105263 2	43.684210526315795	42.631578947368396
TCAP	48.9473684210526	47.36842105263316	46.3157894736842	47.894736842105 3	47.894736842105 3	49.4736842105263	48.94736842105263	47.36842105263316	46.3157894736842
TMB	51.052631578947405	53.684210526315 8	46.842105263157904	48.9473684210526	53.684210526315 8	49.4736842105263	56.3157894736842	53.684210526315 8	46.842105263157904
TOP	43.1578947368421	45.263157894737368	47.89473684210553	46.842105263157904	48.421052631578895	51.578947368421105	43.684210526315795	46.842105263157904	47.89473684210553
TPIPL	52.1052631578947	55.78947368421 05	46.842105263157904	47.894736842105 3	48.947368421052 6	44.2105263157895	52.1052631578947	55.263157894737368	46.3157894736842
TRUE	49.4736842105263	46.3157894736842	46.842105263157904	45.78947368421 05	45.78947368421 05	47.89473684210553	50.5263157894737	49.4736842105263	46.842105263157904
TTW	45.78947368421 05	50.0	39.473684210526 3	41.052631578947 4	50.0	38.9473684210526	45.263157894737368	50.0	38.9473684210526
TU	48.9473684210526	49.4736842105263	44.210526315789 5	44.210526315789 5	44.736842105263 2	39.4736842105263	48.94736842105263	49.4736842105263	44.736842105263 2
WHA	51.052631578947405	50.5263157894737	47.36842105263316	47.894736842105 3	47.894736842105 3	44.736842105263 2	45.78947368421 05	50.5263157894737	47.36842105263316

Table 67 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	45.78947368421 05	51.052631578947405	43.684210526315795	47.894736842105 3	52.631578947368396	53.684210526315 8	48.9473684210526	46.3157894736842	43.684210526315795
AOT	42.1052631578947 04	43.684210526315795	38.9473684210526	53.1578947368421	50.5263157894737	45.263157894736 8	42.1052631578947 04	44.736842105263 2	38.9473684210526
BA	38.88888888888889	38.88888888888889	38.88888888888889	42.592592592595	35.185185185185205	55.55555555555556	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	40.5263157894737	38.9473684210526	41.578947368421 1	57.894736842105296	46.3157894736842	52.1052631578947	40.5263157894737	40.0	41.578947368421 1
BBL	42.1052631578947 04	43.684210526315795	44.736842105263 2	48.9473684210526	43.1578947368421	53.1578947368421	41.578947368421 1	43.684210526315795	44.736842105263 2
BCP	38.4210526315789	38.9473684210526	38.4210526315789	48.9473684210526	44.2105263157895	43.1578947368421	41.0526315789474	37.368421052631604	38.4210526315789
BDMS	40.0	41.0526315789474	37.368421052631604	51.052631578947405	47.894736842105 3	48.421052631578895	40.0	39.4736842105263	37.368421052631604
BEC	48.9473684210526	50.0	43.684210526315795	54.736842105263 19	48.9473684210526	51.578947368421105	47.3684210526316	48.421052631578895	43.684210526315795
BH	48.421052631578895	44.736842105263 2	49.4736842105263	47.3684210526316	45.263157894737368	48.9473684210526	47.3684210526316	44.2105263157895	50.0
BLA	47.89473684210553	51.578947368421105	45.78947368421 05	51.052631578947405	49.4736842105263	46.842105263157904	48.421052631578895	50.5263157894737	45.78947368421 05
BTS	45.263157894736 8	45.78947368421 05	40.5263157894737	46.842105263157904	50.0	56.3157894736842	45.263157894736 8	46.842105263157904	40.5263157894737
CBG	32.0	38.0	32.0	50.0	60.0	46.0	32.0	32.0	32.0
CENTEL	41.57894736842 11	46.842105263157904	41.578947368421 1	49.4736842105263	47.894736842105 3	51.578947368421105	42.631578947368396	46.3157894736842	41.578947368421 1
CK	38.9473684210526	39.4736842105263	38.9473684210526	47.3684210526316	47.3684210526316	54.736842105263 19	40.0	39.4736842105263	38.9473684210526
CPALL	44.736842105263 2	45.263157894736 8	43.157894736842 1	53.6842105263158	48.421052631578895	45.78947368421 05	44.736842105263 2	43.157894736842 1	43.157894736842 1
CPF	46.3157894736842	48.9473684210526	43.684210526315795	50.5263157894737	45.263157894737368	47.3684210526316	45.78947368421 05	48.9473684210526	43.684210526315795
CPN	47.3684210526316	42.631578947368396	38.4210526315789	48.9473684210526	46.3157894736842	50.0	38.4210526315789	42.631578947368396	38.4210526315789
DELTA	46.842105263157904	47.89473684210553	45.263157894736 8	51.578947368421105	47.3684210526316	52.1052631578947	46.842105263157904	47.3684210526316	45.263157894736 8
DTAC	52.631578947368396	52.631578947368396	50.0	44.736842105263 2	47.894736842105 3	50.0	53.1578947368421	52.631578947368396	50.0
EGCO	41.0526315789474	42.1052631578947 04	38.9473684210526	50.5263157894737	47.3684210526316	46.3157894736842	41.0526315789474	43.1578947368421	38.9473684210526
GLOW	41.5789473684211	40.0	41.578947368421 1	52.631578947368396	50.0	46.3157894736842	41.0526315789474	40.526315789474 7	41.578947368421 1
GPSC	43.75	43.75	43.75	56.25	56.25	43.75	43.75	50.0	43.75
HMPRO	40.526315789474 7	38.9473684210526	36.842105263179	47.368421052631 6	47.368421052631 6	55.78947368421 05	38.4210526315789	39.4736842105263	36.842105263179
INTUCH	42.1052631578947 04	42.631578947368396	42.1052631578947 04	51.052631578947405	48.947368421052 6	52.631578947368396	42.631578947368396	41.0526315789474	42.1052631578947 04
IRPC	46.3157894736842	45.263157894736 8	43.157894736842 1	54.736842105263 19	46.842105263157904	52.631578947368396	46.842105263157904	45.78947368421 05	43.157894736842 1
IVL	47.36842105263316	51.052631578947405	42.631578947368396	46.842105263157904	55.78947368421 05	60.526315789473 69	52.1052631578947	53.1578947368421	42.631578947368396
KBANK	46.3157894736842	43.1578947368421	45.263157894736 8	50.5263157894737	44.2105263157895	46.842105263157904	45.78947368421 05	44.736842105263 2	45.263157894736 8
KCE	44.2105263157895	47.89473684210553	40.0	50.0	52.1052631578947	50.5263157894737	41.578947368421 1	47.36842105263316	40.0
KTB	40.5263157894737	41.578947368421 1	39.473684210526 3	50.5263157894737	48.947368421052 6	45.263157894736 8	40.5263157894737	41.052631578947 4	39.473684210526 3
LH	41.0526315789474	41.578947368421 1	38.4210526315789	46.842105263157904	36.3157894736842	43.1578947368421	40.5263157894737	39.473684210526 3	38.4210526315789
MINT	41.0526315789474	41.052631578947 4	42.1052631578947 04	52.631578947368396	45.263157894736 8	48.421052631578895	42.1052631578947 04	38.4210526315789	42.1052631578947 04
MTLS	51.020408163265294	51.020408163265294	48.9795918367347	34.6938775510204	55.1020408163265	57.142857142857096	51.020408163265294	51.020408163265294	48.9795918367347
PS	37.894736842105296	43.684210526315795	36.3157894736842	55.78947368421 05	50.0	52.631578947368396	37.894736842105296	37.368421052631604	36.3157894736842
PTT	40.5263157894737	50.0	41.052631578947 4	58.421052631578995	46.842105263157904	38.4210526315789	41.052631578947 4	47.89473684210553	41.052631578947 4
PTTEP	51.052631578947405	52.1052631578947	54.210526315789 5	42.1052631578947 04	43.157894736842 1	45.78947368421 05	50.0	53.1578947368421	54.210526315789 5
PTTGC	46.842105263157904	48.9473684210526	46.842105263157904	48.421052631578895	50.5263157894737	46.842105263157904	46.842105263157904	47.36842105263316	46.842105263157904
ROBINS	41.57894736842 11	44.210526315789 5	37.894736842105296	54.736842105263 19	52.631578947368396	52.1052631578947	45.78947368421 05	45.263157894736 8	37.894736842105296
SAWAD	40.0	38.88888888888889	41.111111111111 1	46.666666666667	46.666666666667	44.444444444444 4	38.88888888888889	38.88888888888889	41.111111111111 1
SCB	51.578947368421105	49.4736842105263	46.3157894736842	49.4736842105263	51.052631578947405	50.0	50.5263157894737	54.736842105263 19	46.842105263157904
SCC	40.0	45.78947368421 05	40.0	45.78947368421 05	48.947368421052 6	50.0	40.0	38.9473684210526	40.0
TASCO	46.3157894736842	46.842105263157904	46.3157894736842	53.6842105263158	47.3684210526316	53.6842105263158	46.3157894736842	46.842105263157904	46.3157894736842
TCAP	38.4210526315789	37.368421052631604	38.4210526315789	51.578947368421105	44.2105263157895	46.842105263157904	38.4210526315789	37.368421052631604	38.4210526315789
TMB	42.631578947368396	40.0	43.684210526315795	54.736842105263 19	44.736842105263 2	50.5263157894737	43.684210526315795	41.578947368421 1	43.684210526315795
TOP	43.1578947368421	44.210526315789 5	40.0	43.684210526315795	45.78947368421 05	50.0	43.1578947368421	43.684210526315795	40.0
TPIPL	47.3684210526316	46.3157894736842	44.210526315789 5	47.894736842105 3	50.0	55.263157894737368	47.3684210526316	46.842105263157904	44.210526315789 5
TRUE	43.684210526315795	37.894736842105296	45.263157894736 8	45.263157894736 8	51.052631578947405	50.5263157894737	41.57894736842 11	44.210526315789 5	45.263157894736 8
TTW	34.736842105263 2	42.631578947368396	34.736842105263 2	51.578947368421105	43.684210526315795	50.0	34.736842105263 2	42.631578947368396	34.736842105263 2
TU	40.0	44.736842105263 2	40.0	50.5263157894737	46.842105263157904	52.631578947368396	40.0	45.263157894736 8	40.0
WHA	42.631578947368396	45.263157894736 8	42.631578947368396	45.78947368421 05	52.1052631578947	56.842105263157904	42.1052631578947 04	47.89473684210553	43.157894736842 1

PredictDirection5 : Buy-Sell Volume

Table 68 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.6315789747368396	44.2105263157895	43.1578947368421	43.1578947368421	42.6315789747368396	43.1578947368421	42.6315789747368396	44.7368421052632	43.1578947368421
AOT	59.4736842105263	60.0	58.4210526315789	59.4736842105263	58.9473684210526	58.9473684210526	58.9473684210526	60.0	58.4210526315789
BA	51.8518518518518	53.7037037037035	48.1481481481481	51.8518518518518	50.0	57.4074074074074	53.7037037037035	53.7037037037035	48.1481481481481
BANPU	41.0526315789474	41.0526315789474	42.1052631578947	40.0	41.5789473684211	40.0	41.0526315789474	41.5789473684211	41.5789473684211
BBL	41.5789473684211	43.6842105263157	41.5789473684211	44.7368421052632	43.1578947368421	42.1052631578947	40.5263157894737	43.6842105263157	41.5789473684211
BCP	54.2105263157895	54.2105263157895	58.4210526315789	48.9473684210526	53.6842105263158	56.3157894736842	55.2631578947368	54.2105263157895	56.8421052631579
BDMS	57.8947368421052	57.8947368421052	58.4210526315789	57.3684210526304	57.8947368421052	57.8947368421052	57.8947368421052	57.8947368421052	58.4210526315789
BEC	47.3684210526316	47.3684210526316	44.7368421052632	46.3157894736842	47.3684210526316	45.2631578947368	46.8421052631579	47.8947368421053	44.7368421052632
BH	55.7894736842105	55.2631578947368	47.8947368421053	55.2631578947368	53.6842105263158	47.3684210526316	57.3684210526304	54.7368421052639	48.4210526315789
BLA	48.4210526315789	48.4210526315789	48.4210526315789	47.3684210526316	47.3684210526316	47.8947368421053	48.4210526315789	47.8947368421053	48.4210526315789
BTS	52.6315789747368396	53.6842105263158	51.0526315789474 05	53.1578947368421	53.1578947368421	53.6842105263158	54.2105263157895	53.6842105263158	51.0526315789474 05
CBG	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0
CENTEL	56.842105263157904	56.842105263157904	59.473684210526 3	56.842105263157904	57.368421052631604	58.4210526315789 95	56.842105263157904	57.368421052631604	59.473684210526 3
CK	57.8947368421052 96	54.2105263157895	54.7368421052639	56.3157894736842	56.842105263157904	52.6315789473683 96	54.2105263157895	54.7368421052639	54.7368421052639
CPALL	56.3157894736842	56.842105263157904	54.7368421052639	56.3157894736842	57.368421052631604	56.3157894736842	54.7368421052639	56.842105263157904	54.7368421052639
CPF	47.3684210526316	45.7894736842105	44.2105263157895	45.2631578947368	45.7894736842105	48.4210526315789 95	47.3684210526316	46.3157894736842	44.7368421052632
CPN	60.0	61.5789473684211	60.0	59.4736842105263	61.052631578947405	59.4736842105263	60.0	61.5789473684211	59.4736842105263
DELTA	51.5789473684211 05	54.7368421052639	54.7368421052639	51.0526315789474 05	54.2105263157895	56.3157894736842	53.6842105263158	54.2105263157895	55.2631578947368
DTAC	50.0	51.0526315789474 05	51.5789473684211 05	50.0	50.5263157894737	51.0526315789474 05	50.0	51.5789473684211 05	51.5789473684211 05
EGCO	55.7894736842105	55.2631578947368	56.3157894736842	55.7894736842105	55.2631578947368	57.8947368421052 96	55.7894736842105	56.3157894736842	56.3157894736842
GLOW	52.6315789747368396	50.5263157894737	54.2105263157895	50.5263157894737	52.1052631578947	53.1578947368421	52.1052631578947	50.0	54.7368421052639
GPSC	56.25	56.25	56.25	50.0	50.0	56.25	56.25	56.25	56.25
HMPRO	58.4210526315789 95	56.842105263157904	60.5263157894739	58.4210526315789 95	57.368421052631604	59.4736842105263	56.842105263157904	56.842105263157904	61.052631578947405
INTUCH	41.5789473684211	42.1052631578947 04	41.0526315789474	43.1578947368421	43.684210526315795	41.5789473684211	41.5789473684211	42.1052631578947 04	41.0526315789474
IRPC	53.1578947368421	55.2631578947368	54.2105263157895	53.1578947368421	54.7368421052639	54.2105263157895	53.1578947368421	54.7368421052639	54.2105263157895
IVL	56.3157894736842	56.3157894736842	55.7894736842105	55.7894736842105	56.842105263157904	55.7894736842105	57.8947368421052 96	55.2631578947368	55.7894736842105
KBANK	45.7894736842105	45.7894736842105	45.7894736842105	45.2631578947368	46.842105263157904	46.3157894736842	45.2631578947368	45.7894736842105	45.7894736842105
KCE	60.0	61.052631578947405	60.0	59.4736842105263	61.052631578947405	60.0	59.4736842105263	61.052631578947405	60.0
KTB	47.8947368421053	46.3157894736842	48.4210526315789 5	46.842105263157904	46.842105263157904	50.5263157894737	46.842105263157904	46.3157894736842	48.4210526315789 5
LH	53.1578947368421	52.6315789473683 96	56.3157894736842	50.5263157894737	48.4210526315789 5	55.2631578947368	53.6842105263158	51.5789473684211 05	55.7894736842105
MINT	58.9473684210526	58.4210526315789 5	57.368421052631604	58.4210526315789 5	58.4210526315789 5	57.368421052631604	58.9473684210526	58.9473684210526	57.368421052631604
MTLS	53.0612244897959	57.1428571428570 96	53.0612244897959	59.1836734693878	57.1428571428570 96	55.1020408163265	55.1020408163265	57.1428571428570 96	53.0612244897959
PS	53.6842105263158	55.7894736842105	63.1578947368421	58.9473684210526	55.2631578947368	65.2631578947368	56.842105263157904	55.7894736842105	63.1578947368421
PTT	44.2105263157895	43.6842105263157 95	50.0	42.1052631578947 04	44.2105263157895	49.4736842105263	44.7368421052632	43.1578947368421	50.0
PTTEP	52.6315789747368396	49.4736842105263	45.7894736842105	51.5789473684211 05	49.4736842105263	47.3684210526316	48.9473684210526	49.4736842105263	44.7368421052632
PTTGC	54.7368421052639	56.3157894736842	56.3157894736842	54.2105263157895	55.7894736842105	55.7894736842105	54.2105263157895	56.842105263157904	56.3157894736842
ROBINS	51.5789473684211 05	51.0526315789474 05	45.7894736842105	48.9473684210526	50.5263157894737	46.842105263157904	51.0526315789474 05	51.5789473684211 05	46.3157894736842
SAWAD	42.2222222222222	42.2222222222222	47.7777777777778	45.5555555555556	42.2222222222222	43.3333333333333	43.3333333333333	44.4444444444444	47.7777777777778
SCB	42.6315789747368396	41.5789473684211	43.684210526315795	44.2105263157895	42.1052631578947 04	43.684210526315795	42.6315789747368396	42.1052631578947 04	43.684210526315795
SCC	48.4210526315789 5	51.0526315789474 05	51.0526315789474 05	47.8947368421053	48.9473684210526	51.0526315789474 05	47.8947368421053	51.0526315789474 05	50.5263157894737
TASCO	50.5263157894737	52.6315789473683 96	49.4736842105263	52.1052631578947	56.842105263157904	50.5263157894737	51.0526315789474 05	52.1052631578947	49.4736842105263
TCAP	53.6842105263158	54.2105263157895	54.7368421052639	55.2631578947368	54.7368421052639	56.3157894736842	53.6842105263158	54.7368421052639	54.7368421052639
TMB	56.842105263157904	56.842105263157904	52.6315789473683 96	55.2631578947368	56.842105263157904	56.3157894736842	57.368421052631604	55.7894736842105	52.6315789473683 96
TOP	59.4736842105263	60.0	60.0	60.0	60.0	59.4736842105263	59.4736842105263	60.0	60.0
TPIPL	54.2105263157895	57.368421052631604	47.3684210526316	56.842105263157904	55.2631578947368	52.6315789473683 96	56.3157894736842	57.368421052631604	47.8947368421053
TRUE	46.842105263157904	47.8947368421053	51.5789473684211 05	45.2631578947368	49.4736842105263	49.4736842105263	52.6315789473683 96	48.4210526315789 5	51.5789473684211 05
TTW	59.4736842105263	56.842105263157904	63.6842105263158	54.7368421052639	57.368421052631604	63.1578947368421	54.2105263157895	57.368421052631604	64.7368421052632
TU	54.2105263157895	50.5263157894737	52.1052631578947	47.8947368421053	47.3684210526316	53.1578947368421	54.2105263157895	51.0526315789474 05	52.1052631578947
WHA	56.842105263157904	59.4736842105263	58.4210526315789 95	56.3157894736842	56.3157894736842	57.8947368421052 96	57.8947368421052 96	59.4736842105263	57.8947368421052 96

Table 69 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	50.0	53.6842105263158	51.5789473684211 05	46.842105263157904	48.4210526315789 5	55.2631578947368	51.0526315789474 05	53.1578947368421	52.1052631578947
AOT	46.842105263157904	47.3684210526316	43.1578947368421	48.4210526315789 5	44.7368421052632	42.1052631578947 04	45.7894736842105	45.2631578947368	43.684210526315795
BA	29.6296296296296	27.7777777777778	31.4814814814815	55.5555555555556	51.8518518518505	46.2962962962963 05	29.6296296296296	31.4814814814815	31.4814814814815
BANPU	52.1052631578947	52.1052631578947	52.6315789473683 96	41.0526315789474	45.2631578947368	47.3684210526316	51.5789473684211 05	53.6842105263158	52.6315789473683 96
BBL	43.684210526315795	46.842105263157904	45.2631578947368	38.4210526315789	42.6315789473683 96	41.5789473684211	42.1052631578947 04	45.2631578947368	45.2631578947368
BCP	46.842105263157904	47.3684210526316	50.5263157894737	44.7368421052632	49.4736842105263	43.684210526315795	49.4736842105263	46.842105263157904	50.5263157894737
BDMS	43.1578947368421	44.7368421052632	42.1052631578947 04	51.5789473684211 05	48.9473684210526	44.7368421052632	44.2105263157895	44.2105263157895	42.1052631578947 04
BEC	48.9473684210526	51.0526315789474 05	50.0	47.8947368421053	50.0	48.4210526315789 5	50.5263157894737	51.5789473684211 05	50.5263157894737
BH	50.5263157894737	48.4210526315789 5	47.3684210526316	50.5263157894737	50.0	48.9473684210526	50.5263157894737	48.4210526315789 5	46.8421052631579 04
BLA	53.6842105263158	45.7894736842105	42.6315789473683 96	50.5263157894737	53.6842105263158	52.1052631578947	52.1052631578947	49.4736842105263	42.1052631578947 04
BTS	51.5789473684211 05	48.4210526315789 5	48.9473684210526	48.4210526315789 5	45.2631578947368	39.4736842105263	52.6315789473683 96	48.9473684210526	48.9473684210526
CBG	42.0	44.0	34.0	54.0	57.999999999999	57.999999999999	40.0	42.0	38.0
CENTEL	45.2631578947368	47.8947368421053	43.1578947368421	47.3684210526316	48.9473684210526	43.684210526315795	46.842105263157904	47.8947368421053	42.6315789473683 96
CK	51.0526315789474 05	49.4736842105263	45.7894736842105	41.5789473684211	44.2105263157895	43.1578947368421	50.5263157894737	50.0	45.7894736842105
CPALL	53.6842105263158	54.2105263157895	51.0526315789474 05	54.2105263157895	52.6315789473683 96	48.9473684210526	53.1578947368421	53.6842105263158	51.5789473684211 05
CPF	52.1052631578947	52.1052631578947	55.7894736842105	45.7894736842105	46.3157894736842	48.4210526315789 5	51.0526315789474 05	53.1578947368421	56.3157894736842
CPN	51.5789473684211 05	54.2105263157895	58.4210526315789 95	44.2105263157895	44.2105263157895	38.4210526315789	55.2631578947368	53.6842105263158	58.4210526315789 95
DELTA	55.2631578947368	51.5789473684211 05	50.0	51.5789473684211 05	50.0	44.2105263157895	53.6842105263158	52.1052631578947	50.0
DTAC	54.7368421052639	52.6315789473683 96	52.6315789473683 96	53.6842105263158	51.0526315789474 05	51.0526315789474 05	54.2105263157895	53.1578947368421	52.1052631578947
EGCO	45.7894736842105	44.7368421052632	44.7368421052632	43.684210526315795	45.7894736842105	47.3684210526316	48.4210526315789 5	43.684210526315795	44.7368421052632
GLOW	51.5789473684211 05	51.0526315789474 05	50.0	45.7894736842105	44.2105263157895	51.0526315789474 05	51.5789473684211 05	52.1052631578947	50.0
GPSC	56.25	56.25	50.0	43.75	43.75	43.75	56.25	56.25	50.0
HMPRO	55.2631578947368	57.8947368421052 96	60.0	49.4736842105263	49.4736842105263	48.9473684210526	53.6842105263158	57.8947368421052 96	60.526315789473 69
INTUCH	52.1052631578947	54.2105263157895	47.3684210526316	52.1052631578947	51.0526315789474 05	53.1578947368421	49.4736842105263	54.2105263157895	48.4210526315789 5
IRPC	47.3684210526316	52.6315789473683 96	47.8947368421053	48.4210526315789 5	50.0	48.9473684210526	49.4736842105263	51.5789473684211 05	47.8947368421053
IVL	52.1052631578947	54.2105263157895	53.1578947368421	52.6315789473683 96	53.1578947368421	47.8947368421053	50.5263157894737	54.7368421052639	53.1578947368421
KBANK	45.7894736842105	45.7894736842105	44.2105263157895	52.1052631578947	45.7894736842105	42.6315789473683 96	44.2105263157895	46.3157894736842	44.2105263157895
KCE	54.7368421052639	51.5789473684211 05	54.2105263157895	49.4736842105263	46.842105263157904	43.684210526315795	51.0526315789474 05	54.2105263157895	54.2105263157895
KTB	50.0	50.5263157894737	48.4210526315789 5	46.842105263157904	46.3157894736842	41.0526315789474	48.9473684210526	50.0	48.4210526315789 5
LH	47.3684210526316	46.842105263157904	43.1578947368421	42.6315789473683 96	42.6315789473683 96	41.5789473684211	46.842105263157904	46.842105263157904	43.684210526315795
MINT	47.8947368421053	47.8947368421053	47.3684210526316	45.7894736842105	45.2631578947368	48.9473684210526	47.8947368421053	47.3684210526316	46.842105263157904
MTLS	44.8979591836735	48.9795918367347	44.8979591836735	51.1020408163265	51.0612244897959	51.020408163265 29	42.8571428571429	51.020408163265 294	44.8979591836735
PS	48.9473684210526	50.5263157894737	45.7894736842105	51.5789473684211 05	51.5789473684211 05	52.1052631578947	50.5263157894737	50.5263157894737	45.7894736842105
PTT	53.6842105263158	50.5263157894737	48.9473684210526	46.842105263157904	45.7894736842105	44.2105263157895	54.2105263157895	49.4736842105263	48.9473684210526
PTTEP	58.4210526315789 95	56.842105263157904	53.1578947368421	54.2105263157895	56.3157894736842	44.7368421052632	58.4210526315789 95	56.842105263157904	53.6842105263158
PTTGC	51.0526315789474 05	54.2105263157895	48.4210526315789 5	55.7894736842105	57.8947368421052 96	57.368421052631604	51.0526315789474 05	53.1578947368421	48.4210526315789 5
ROBINS	55.2631578947368	55.7894736842105	48.4210526315789 5	46.3157894736842	49.4736842105263	48.4210526315789 5	55.7894736842105	55.2631578947368	48.4210526315789 5
SAWAD	52.2222222222222	56.6666666666667	53.3333333333333	45.5555555555556	46.6666666666667	53.3333333333333	54.4444444444439	53.3333333333333	
SCB	56.3157894736842	56.842105263157904	52.6315789473683 96	48.9473684210526	47.8947368421053	46.3157894736842	56.3157894736842	55.2631578947368	52.1052631578947
SCC	48.9473684210526	47.3684210526316	46.842105263157904	43.1578947368421	44.2105263157895	41.5789473684211	46.842105263157904	47.3684210526316	46.3157894736842
TASCO	50.0	46.3157894736842	42.1052631578947 04	44.7368421052632	43.684210526315795	46.3157894736842	50.0	46.3157894736842	42.1052631578947 04
TCAP	45.2631578947368	46.3157894736842	42.6315789473683 96	48.4210526315789 5	48.9473684210526	53.6842105263158	45.7894736842105	46.3157894736842	42.6315789473683 96
TMB	49.4736842105263	46.3157894736842	47.3684210526316	48.4210526315789 5	50.0	47.3684210526316	48.4210526315789 5	46.842105263157904	47.3684210526316
TOP	41.0526315789474	43.684210526315795	41.0526315789474	44.2105263157895	46.3157894736842	47.3684210526316	40.5263157894737	43.684210526315795	41.0526315789474
TPIPL	45.7894736842105	53.1578947368421	45.2631578947368	51.5789473684211 05	50.5263157894737	46.3157894736842	54.2105263157895	52.6315789473683 96	45.2631578947368
TRUE	48.4210526315789 5	48.4210526315789 5	51.5789473684211 05	46.3157894736842	44.7368421052632	47.3684210526316	48.9473684210526	48.9473684210526	51.5789473684211 05
TTW	42.1052631578947 04	44.7368421052632	40.0	41.0526315789474	42.1052631578947 04	45.7894736842105	42.6315789473683 96	44.7368421052632	39.4736842105263
TU	46.842105263157904	46.3157894736842	46.842105263157904	36.3157894736842	37.894736842105296	40.0	47.3684210526316	45.7894736842105	46.3157894736842
WHA	49.4736842105263	46.3157894736842	52.6315789473683 96	57.368421052631604	55.7894736842105	55.2631578947368	50.5263157894737	45.7894736842105	53.1578947368421

Table 70 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	52.631578947368396	51.578947368421105	51.578947368421105	54.2105263157895	54.2105263157895	55.26315789473684	50.5263157894737	52.631578947368396	51.578947368421105
AOT	49.473684210526	48.9473684210526	47.8947368421053	47.8947368421053	45.7894736842105	42.631578947368396	48.9473684210526	47.3684210526316	48.421052631578895
BA	33.33333333333333	35.185185185185205	35.185185185185205	57.4074074074074	53.703703703703695	38.88888888888889	33.33333333333333	35.185185185185205	33.33333333333333
BANPU	51.578947368421105	53.68421052631578	51.578947368421105	51.052631578947405	45.26315789473684	46.842105263157904	52.1052631578947	53.1578947368421	51.052631578947405
BBL	44.21052631578947	45.26315789473684	44.73684210526316	38.94736842105263	40.52631578947368	37.36842105263164	44.21052631578947	44.736842105263	45.26315789473684
BCP	49.473684210526	47.8947368421053	50.52631578947368	48.94736842105263	49.473684210526	43.684210526315795	46.842105263157904	46.842105263157904	50.52631578947368
BDMS	45.78947368421105	47.36842105263164	42.105263157894704	51.052631578947405	47.36842105263164	44.73684210526316	43.157894736842	46.31578947368421	42.105263157894704
BEC	51.578947368421105	50.52631578947368	49.473684210526	52.631578947368396	49.473684210526	49.473684210526	51.052631578947405	50.52631578947368	50.0
BH	47.89473684210526	50.52631578947368	48.421052631578895	50.0	51.052631578947405	48.421052631578895	50.0	50.0	48.421052631578895
BLA	53.1578947368421	46.842105263157904	42.631578947368396	51.052631578947405	52.1052631578947	52.1052631578947	53.1578947368421	45.78947368421105	42.631578947368396
BTS	52.10526315789474	48.94736842105263	50.52631578947368	43.684210526315795	42.631578947368396	42.631578947368396	52.1052631578947	48.94736842105263	50.0
CBG	44.0	50.0	34.0	52.0	54.0	57.99999999999999	44.0	42.0	38.0
CENTEL	45.78947368421105	46.31578947368421	42.105263157894704	49.473684210526	47.36842105263164	45.26315789473684	45.26315789473684	46.31578947368421	42.105263157894704
CK	46.842105263157904	48.421052631578895	46.842105263157904	52.1052631578947	44.2105263157895	46.31578947368421	46.842105263157904	48.94736842105263	46.31578947368421
CPALL	52.10526315789474	53.684210526315	51.052631578947405	55.78947368421105	53.157894736842105	48.94736842105263	53.157894736842105	53.684210526315	51.578947368421105
CPF	53.684210526315	52.631578947368396	57.368421052631604	55.789473684210	51.578947368421105	53.684210526315	56.842105263157904	51.052631578947405	57.368421052631604
CPN	54.21052631578895	53.684210526315	58.421052631578995	45.789473684210	43.684210526315795	38.421052631579	55.789473684210	52.1052631578947	58.421052631578995
DELTA	53.1578947368421	50.0	50.0	51.578947368421105	51.578947368421105	44.21052631578895	53.684210526315	52.1052631578947	50.0
DTAC	50.0	52.1052631578947	52.631578947368396	50.5263157894737	51.052631578947405	53.684210526315	50.0	53.1578947368421	52.631578947368396
EGCO	46.842105263157904	45.26315789473684	44.73684210526316	43.684210526315795	46.315789476842	47.368421052631	47.368421052631	44.2105263157895	44.736842105263
GLOW	51.578947368421105	50.0	50.5263157894737	43.684210526315795	44.2105263157895	50.5263157894737	52.1052631578947	50.5263157894737	50.0
GPSC	56.25	62.5	50.0	37.5	43.75	43.75	56.25	56.25	50.0
HMPRO	55.789473684105	56.842105263157904	59.473684210263	45.263157894736	48.421052631578895	48.421052631578895	54.73684210526319	57.368421052631604	59.473684210263
INTUCH	50.52631578947368	53.1578947368421	47.3684210526316	52.1052631578947	50.0	53.157894736842	48.9473684210526	52.631578947368396	48.421052631578895
IRPC	48.421052631578895	52.631578947368396	47.3684210526316	47.8947368421053	51.578947368421105	45.789473684210	48.9473684210526	50.52631578947368	47.3684210526316
IVL	52.10526315789474	50.52631578947368	51.052631578947405	46.315789476842	46.315789476842	49.473684210263	50.52631578947368	49.473684210263	51.052631578947405
KBANK	45.789473684105	44.2105263157895	46.315789476842	44.736842105263	42.631578947368396	47.368421052631	47.368421052631	44.2105263157895	46.315789476842
KCE	54.21052631578895	51.578947368421105	54.2105263157895	48.421052631578895	46.842105263157904	41.052631578947	53.157894736842	52.1052631578947	53.157894736421
KTB	48.421052631578895	50.52631578947368	48.421052631578895	46.842105263157904	46.842105263157904	41.052631578947	48.947368421026	50.0	48.421052631578895
LH	48.94736842105263	50.52631578947368	46.842105263157904	52.1052631578947	50.52631578947368	44.736842105263	48.947368421026	50.0	46.842105263157904
MINT	49.473684210263	46.842105263157904	43.157894736842	46.842105263157904	46.315789476842	48.421052631578895	47.894736842103	46.842105263157904	43.157894736842
MTLS	44.8979591836735	48.9795918367347	44.8979591836735	55.102040816326	55.102040816326	51.020408163265294	44.8979591836735	46.93877551020141	44.8979591836735
PS	50.52631578947368	51.578947368421105	46.842105263157904	47.368421052632	45.26315789473684	45.78947368421105	51.052631578947405	51.578947368421105	46.315789476842
PTT	53.684210526315	50.0	48.94736842105263	48.94736842105263	45.78947368421105	44.2105263157895	53.684210526315	49.473684210263	48.94736842105263
PTTEP	62.1052631578947	58.947368421052	53.684210526315	51.578947368421105	45.26315789473684	46.842105263157904	61.578947368211	59.473684210263	53.157894736421
PTTGC	49.473684210263	53.684210526315	48.421052631578895	55.789473684210	58.421052631578995	57.368421052631604	50.52631578947368	53.684210526315	48.421052631578895
ROBINS	55.263157894736	55.263157894736	48.421052631578895	44.736842105263	49.473684210263	48.421052631578895	55.263157894736	55.263157894736	48.421052631578895
SAWAD	54.444444444444439	56.666666666667	53.333333333333	46.666666666667	45.555555555556	46.666666666667	55.555555555556	55.555555555556	53.333333333333
SCB	56.31578947368421	55.789473684105	52.631578947368396	47.894736842053	47.894736842053	46.315789476842	56.842105263157904	55.789473684105	52.1052631578947
SCC	50.0	47.894736842053	47.894736842053	40.526315789737	43.157894736842	38.421052631579	49.473684210263	48.421052631578895	47.894736842053
TASCO	49.473684210263	52.631578947368396	43.684210526315795	45.789473684210	46.315789476842	46.315789476842	52.1052631578947	51.578947368421105	43.684210526315795
TCAP	46.315789476842	46.315789476842	42.631578947368396	48.421052631578895	48.947368421026	53.684210526315	46.315789476842	45.789473684210	42.631578947368396
TMB	48.947368421026	47.368421052631	47.368421052631	46.842105263157904	51.578947368421105	46.842105263157904	48.421052631578895	47.894736842053	47.368421052631
TOP	43.684210526315795	44.2105263157895	41.052631578947	44.736842105263	45.789473684210	47.368421052631	42.631578947368396	42.631578947368396	41.052631578947
TPIPL	46.842105263157904	53.157894736842	44.2105263157895	47.894736842053	51.578947368421105	45.789473684210	45.263157894736	53.157894736842	44.2105263157895
TRUE	52.1052631578947	51.052631578947405	47.368421052631	46.842105263157904	46.842105263157904	46.842105263157904	53.684210526315	51.052631578947405	46.842105263157904
TTW	42.105263157894704	43.684210526315795	41.578947368421	43.684210526315795	43.157894736842	44.2105263157895	40.526315789737	43.157894736842	41.052631578947
TU	45.789473684105	45.789473684105	46.842105263157904	36.842105263179	37.368421052631604	40.0	46.842105263157904	45.789473684105	46.315789476842
WHA	53.684210526158	47.894736842053	51.578947368421105	55.263157894736	57.368421052631604	56.842105263157904	52.1052631578947	47.368421052631	51.578947368421105

Table 71 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.1578947368421	45.26315789473684	43.684210526315795	48.947368421026	51.578947368421105	51.578947368421105	46.842105263157904	45.26315789473684	43.684210526315795
AOT	41.5789473684211	45.78947368421105	40.526315789737	50.526315789737	47.894736842053	52.1052631578947	42.1052631578947	44.2105263157895	40.526315789737
BA	31.481481481815	31.481481481815	38.88888888888889	48.148148148141	37.037037037037	42.592592592595	31.481481481815	31.481481481815	38.88888888888889
BANPU	40.0	39.473684210263	41.578947368211	48.947368421026	46.842105263157904	55.263157894368	40.0	40.0	41.578947368211
BBL	43.1578947368421	40.0	44.73684210526316	52.631578947368396	45.263157894736	46.842105263157904	43.684210526315795	40.526315789737	44.736842105263
BCP	40.526315789737	40.526315789737	38.421052631579	53.684210526315	48.421052631578895	52.1052631578947	40.526315789737	42.631578947368396	38.421052631579
BDMS	40.526315789737	39.473684210263	38.421052631579	45.789473684210	41.052631578974	43.157894736421	40.526315789737	39.473684210263	38.421052631579
BEC	46.315789476842	48.421052631578895	45.263157894736	50.0	47.894736842053	51.578947368421105	47.894736842053	47.894736842053	45.263157894736
BH	47.368421052631	51.578947368421105	49.473684210263	52.1052631578947	49.473684210263	56.842105263157904	51.052631578947405	53.684210526315	49.473684210263
BLA	49.473684210263	50.526315789737	46.315789476842	48.421052631578895	51.578947368421105	49.473684210263	48.947368421026	52.1052631578947	45.789473684210
BTS	42.105263157894704	43.684210526315795	40.526315789737	48.421052631578895	46.315789476842	46.315789476842	42.105263157894704	43.684210526315795	40.526315789737
CBG	36.0	36.0	32.0	42.0	42.0	52.0	34.0	32.0	34.0
CENTEL	42.631578947368396	43.157894736421	41.578947368421	46.842105263157904	49.473684210263	50.0	42.631578947368396	43.157894736421	41.578947368421
CK	41.578947368211	42.631578947368396	38.947368210526	44.736842105263	50.526315789737	50.526315789737	50.526315789737	40.526315789737	38.947368210526
CPALL	43.684210526315795	45.26315789473684	43.157894736421	45.263157894736	53.684210526315	52.631578947368396	44.2105263157895	43.157894736421	43.157894736421
CPF	43.684210526315795	42.105263157894704	43.157894736421	49.473684210263	51.578947368421105	45.789473684105	43.684210526315795	41.052631578974	43.157894736421
CPN	40.0	39.473684210263	39.473684210263	42.105263157894704	46.842105263157904	52.1052631578947	38.421052631579	41.052631578974	39.473684210263
DELTA	48.421052631578895	50.526315789737	47.368421052631	45.789473684105	51.052631578947405	51.052631578947405	47.894736842053	50.526315789737	47.368421052631
DTAC	46.842105263157904	50.0	45.263157894736	50.526315789737	52.631578947368396	47.368421052631	48.421052631578895	48.421052631578895	44.736842105263
EGCO	38.947368420526	40.0	38.947368420526	43.157894736421	44.2105263157895	57.894736842105296	39.473684210263	39.473684210263	38.947368420526
GLOW	43.684210526315795	43.684210526315795	40.526315789737	46.315789476842	52.631578947368396	46.842105263157904	42.631578947368396	42.631578947368396	40.526315789737
GPSC	43.75	43.75	43.75	56.25	43.75	50.0	43.75	43.75	43.75
HMPRO	36.315789476842	35.789473684105	36.842105263179	47.894736842053	48.421052631578895	54.2105263157895	35.789473684105	35.789473684105	36.842105263179
INTUCH	42.105263157894704	43.684210526315795	42.105263157894704	50.0	51.052631578947405	53.684210526315	42.631578947368396	43.684210526315795	42.105263157894704
IRPC	43.157894736421	42.105263157894704	43.157894736421	45.789473684105	45.789473684105	54.2105263157895	41.578947368421	42.105263157894704	43.157894736421
IVL	48.421052631578895	50.0	44.2105263157895	44.736842105263	47.894736842053	52.631578947368396	49.473684210263	53.157894736421	44.2105263157895
KBANK	46.842105263157904	42.105263157894704	45.263157894736	44.736842105263	40.0	50.526315789737	46.315789476842	42.105263157894704	45.263157894736
KCE	45.263157894736	46.842105263157904	40.0	53.157894736421	49.473684210263	52.105263157894	45.789473684105	46.315789476842	40.0
KTB	39.473684210263	41.578947368211	39.473684210263	53.157894736421	44.736842105263	47.894736842053	41.578947368211	40.0	39.473684210263
LH	41.052631578974	41.578947368211	38.421052631579	54.736842105263 19	52.105263157894	47.368421052631	41.052631578974	42.105263157894704	38.421052631579
MINT	42.631578947368396	43.157894736421	42.105263157894704	48.947368421026	43.684210526315795	57.368421052631604	41.578947368421	43.157894736421	42.105263157894704
MTLS	53.061224497959	48.9795918367347	48.9795918367347	44.8979591836735	48.979591836735 47	51.020408163265294	51.020408163265294	51.020408163265294	48.9795918367347
PS	38.421052631579	37.894736842105296	36.315789476842	43.157894736421	44.736842105263 2	46.315789476842	38.421052631579	37.894736842105296	36.315789476842
PTT	51.578947368421105	50.0	43.684210526315795	53.684210526315	47.368421052631 6	47.894736842053	47.894736842053	48.947368421026	44.736842105263 2
PTTEP	56.842105263157904	56.842105263157904	54.2105263157895	47.894736842053	58.947368421026	52.1052631578947	59.473684210263	58.421052631578995	54.2105263157895
PTTGC	41.57894736842 11	43.684210526315795	45.78947368421 05	53.684210526315	43.157894736421	53.157894736421	46.315789476842	45.263157894736	45.78947368421 05
ROBINS	46.842105263157904	44.736842105263 2	38.947368420526	47.368421052631 6	54.2105263157895	42.105263157894704	45.263157894736	44.736842105263 2	40.0
SAWAD	46.666666666667	41.111111111111 11	41.111111111111 11	45.555555555556	50.0	47.777777777778	46.666666666667	41.111111111111 11	41.111111111111 11
SCB	48.947368420526	48.421052631578895	42.105263157894704	49.473684210263	48.421052631578895	50.0	48.421052631578895	50.526315789737	42.631578947368396
SCC	38.947368420526	41.578947368421 1	40.0	46.315789476842	47.368421052631 6	40.0	40.0	42.105263157894704	40.0
TASCO	46.842105263157904	46.842105263157904	46.315789476842	43.157894736421	51.052631578947405	51.052631578947405	47.894736842053	47.894736842053	46.315789476842
TCAP	37.368421052631604	36.842105263179	38.421052631579	46.842105263179	41.578947368421 1	52.631578947368396	38.947368420526	35.789473684105	38.421052631579
TMB	43.684210526315795	43.684210526315795	43.684210526315795	46.315789476842	44.2105263157895	46.842105263157904	43.684210526315795	43.684210526315795	43.684210526315795
TOP	39.473684210263	39.473684210263	40.0	53.157894736421	45.263157894736 8	50.0	40.0	40.526315789737	40.0
TPIPL	44.2105263157895	44.2105263157895	44.2105263157895	51.052631578947405	46.842105263157904	54.736842105263 19	42.105263157894704	41.578947368421 1	44.736842105263 2
TRUE	44.736842105263 2	45.263157894736 8	44.736842105263 2	46.842105263157904	46.315789476842	52.631578947368396	43.684210526315795	44.2105263157895	44.736842105263 2
TTW	33.157894736842096	32.631578947368 4	34.736842105263 2	48.421052631578895	43.157894736421	47.894736842053	33.157894736842096	33.684210526315 8	34.736842105263 2
TU	40.526315789737	39.473684210263	36.315789476842	50.526315789737	40.526315789737	45.263157894736	38.421052631579	38.947368420526	36.315789476842
WHA	44.2105263157895	42.631578947368396	42.105263157894704	38.421052631579	44.2105263157895	42.631578947368396	45.263157894736	43.684210526315795	42.105263157894704

PredictDirection6 : CMF (Chaikin Money Flow)

Table 72 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.21052631578895	44.2105263157895	43.1578947368421	44.21052631578895	43.684210526315795	43.684210526315795	43.1578947368421	43.684210526315795	43.1578947368421
AOT	56.3157894736842	58.421052631578995	59.4736842105263	55.2631578947368	56.842105263157904	59.4736842105263	56.3157894736842	58.421052631578995	59.4736842105263
BA	46.296296296296305	46.296296296296305	44.4444444444444	44.4444444444444	44.4444444444444	48.1481481481481	55.55555555555556	44.4444444444444	46.296296296296305
BANPU	39.4736842105263	39.4736842105263	41.0526315789474	40.0	39.4736842105263	41.0526315789474	40.5263157894737	39.4736842105263	41.0526315789474
BBL	43.684210526315795	42.631578947368396	43.684210526315795	43.684210526315795	43.684210526315795	44.7368421052632	43.684210526315795	40.52631578947368	43.684210526315795
BCP	49.4736842105263	47.8947368421053	53.1578947368421	45.2631578947368	46.3157894736842	55.2631578947368	49.4736842105263	48.9473684210526	53.1578947368421
BDMS	60.0	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	57.368421052631604	58.421052631578995	58.421052631578995
BEC	45.7894736842105	44.2105263157895	45.2631578947368	44.7368421052632	44.2105263157895	45.2631578947368	45.7894736842105	44.2105263157895	45.2631578947368
BH	55.2631578947368	55.2631578947368	53.1578947368421	55.2631578947368	52.1052631578947	52.1052631578947	55.7894736842105	54.2105263157895	53.1578947368421
BLA	47.8947368421053	50.5263157894737	48.421052631578895	47.8947368421053	50.5263157894737	47.8947368421053	49.4736842105263	50.0	48.421052631578895
BTS	54.73684210526319	54.2105263157895	56.3157894736842	55.2631578947368	54.2105263157895	56.842105263157904	55.2631578947368	55.7894736842105	56.3157894736842
CBG	70.0	68.0	68.0	68.0	68.0	68.0	70.0	70.0	68.0
CENTEL	56.3157894736842	57.368421052631604	55.7894736842105	55.2631578947368	57.89473684210596	55.2631578947368	56.3157894736842	58.421052631578995	55.7894736842105
CK	56.842105263157904	57.894736842105296	54.2105263157895	54.2105263157895	58.9473684210526	55.7894736842105	54.2105263157895	57.894736842105296	54.73684210526319
CPALL	52.1052631578947	53.684210526315	56.842105263157904	54.2105263157895	54.2105263157895	56.3157894736842	51.578947368421105	54.2105263157895	56.842105263157904
CPF	48.9473684210526	48.9473684210526	51.578947368421105	48.9473684210526	48.9473684210526	51.578947368421105	50.5263157894737	48.421052631578895	50.5263157894737
CPN	57.894736842105296	57.368421052631604	57.368421052631604	56.3157894736842	57.368421052631604	56.842105263157904	58.9473684210526	57.368421052631604	57.894736842105296
DELTA	53.1578947368421	51.578947368421105	54.73684210526319	55.2631578947368	52.1052631578947	55.7894736842105	54.73684210526319	51.578947368421105	54.73684210526319
DTAC	50.5263157894737	50.5263157894737	51.052631578947405	50.5263157894737	50.5263157894737	51.052631578947405	50.5263157894737	50.5263157894737	51.052631578947405
EGCO	56.3157894736842	56.842105263157904	55.2631578947368	56.842105263157904	56.842105263157904	55.7894736842105	56.3157894736842	56.842105263157904	55.2631578947368
GLOW	50.5263157894737	51.052631578947405	50.5263157894737	51.052631578947405	53.1578947368421	54.2105263157895	50.0	51.052631578947405	50.5263157894737
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	57.894736842105296	60.0	60.0	58.421052631578995	60.526315789473695	60.526315789473695	58.421052631578995	59.473684210526304	60.0
INTUCH	41.0526315789474	43.1578947368421	41.5789473684211	42.1052631578947 04	42.631578947368396	41.5789473684211	41.0526315789474	43.1578947368421	41.5789473684211
IRPC	54.73684210526319	55.7894736842105	54.2105263157895	54.73684210526319	55.7894736842105	54.73684210526319	54.73684210526319	55.7894736842105	54.2105263157895
IVL	56.842105263157904	55.2631578947368	54.73684210526319	54.2105263157895	55.7894736842105	55.7894736842105	54.73684210526319	54.73684210526319	54.73684210526319
KBANK	45.7894736842105	43.1578947368421	44.73684210526 32	44.73684210526 32	46.3157894736842	46.3157894736842	45.2631578947368	45.7894736842105	44.73684210526 32
KCE	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
KTB	49.4736842105263	46.842105263157904	53.68421052631 58	48.421052631578895	47.8947368421053	52.631578947368396	49.4736842105263	47.36842105263 16	53.68421052631 58
LH	47.36842105263 16	45.2631578947368	53.68421052631 58	45.2631578947368	45.7894736842105	54.2105263157895	46.3157894736842	45.2631578947368	53.68421052631 58
MINT	58.421052631578995	58.9473684210526	58.9473684210526	58.421052631578995	58.9473684210526	58.421052631578995	58.9473684210526	58.9473684210526	58.9473684210526
MTLS	55.1020408163265	51.020408163265294	55.1020408163265	48.9795918367347	51.020408163265294	48.9795918367347	55.1020408163265	51.020408163265294	55.1020408163265
PS	53.1578947368421	51.578947368421105	61.052631578947405	52.631578947368396	53.1578947368421	60.526315789473695	54.2105263157895	51.052631578947405	61.57894736842111
PTT	41.052631578947 4	42.105263157894704	43.684210526315795	41.578947368421 1	42.105263157894704	46.3157894736842	41.578947368421 2	42.631578947368396	43.684210526315795
PTTEP	52.1052631578947	46.3157894736842	46.3157894736842	45.7894736842105	48.9473684210526	46.3157894736842	47.3684210526316	46.3157894736842	46.3157894736842
PTTGC	52.631578947368396	54.2105263157895	53.6842105263158	53.6842105263158	53.6842105263158	53.1578947368421	52.1052631578947	54.736842105263 19	53.68421052631 58
ROBINS	50.0	50.5263157894737	53.6842105263158	52.1052631578947	51.578947368421105	51.578947368421105	52.1052631578947	50.0	53.68421052631 58
SAWAD	44.4444444444444	50.0	41.1111111111111	52.2222222222222	46.6666666666667	40.0	45.5555555555556	47.7777777777778	41.1111111111111
SCB	41.0526315789474	41.5789473684211	41.0526315789474	41.5789473684211	41.5789473684211	40.5263157894737	40.5263157894737	41.5789473684211	41.0526315789474
SCC	41.5789473684211	42.631578947368396	45.7894736842105	40.0	42.631578947368396	45.2631578947368	45.2631578947368	43.684210526315795	45.7894736842105
TASCO	50.0	51.052631578947405	54.73684210526319	50.5263157894737	51.578947368421105	54.73684210526319	50.5263157894737	51.052631578947405	54.2105263157895
TCAP	51.052631578947405	50.5263157894737	53.6842105263158	51.052631578947405	52.1052631578947	52.631578947368396	51.052631578947405	50.0	53.68421052631 58
TMB	56.3157894736842	56.3157894736842	55.7894736842105	58.9473684210526	56.842105263157904	57.368421052631604	56.3157894736842	54.2105263157895	55.7894736842105
TOP	60.0	59.473684210526 3	60.0	59.473684210526 3	58.9473684210526	59.473684210526 3	59.473684210526 3	59.473684210526 3	60.0
TPIPL	57.368421052631604	51.578947368421105	56.842105263157904	54.2105263157895	52.631578947368396	56.3157894736842	59.473684210526 3	52.1052631578947	56.842105263157904
TRUE	50.5263157894737	45.2631578947368	49.473684210526 3	46.842105263157904	47.368421052631 6	49.473684210526 3	50.5263157894737	45.2631578947368	49.473684210526 3
TTW	47.8947368421053	50.0	51.052631578947405	42.631578947368396	49.473684210526 3	55.2631578947368	47.368421052631 6	50.0	51.052631578947405
TU	55.7894736842105	53.684210526315 8	56.842105263157904	55.2631578947368	52.631578947368396	54.73684210526319	53.684210526315 8	53.684210526315 8	56.842105263157904
WHA	54.73684210526319	53.1578947368421	60.0	54.73684210526319	52.1052631578947	57.368421052631604	53.684210526315 8	53.1578947368421	60.0

Table 73 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	45.2631578947368	51.052631578947405	43.684210526315795	48.9473684210526	48.9473684210526	41.5789473684211	45.7894736842105	51.052631578947405	43.684210526315795
AOT	46.842105263157904	48.421052631578895	42.631578947368396	45.2631578947368	42.631578947368396	39.4736842105263	43.684210526315795	46.842105263157904	42.631578947368396
BA	42.592592592592595	42.592592592592595	46.296296296296305	57.4074074074074	66.66666666666667	68.5185185185185	40.7407407407407	44.4444444444444	46.296296296296305
BANPU	51.578947368421105	52.631578947368396	56.31578947368 42	47.36842105263 16	50.0	46.3157894736842	51.052631578947405	53.684210526315 8	56.31578947368 42
BBL	53.1578947368421	52.631578947368396	46.3157894736842	53.1578947368421	53.684210526315 8	49.4736842105263	54.73684210526319	53.1578947368421	46.3157894736842
BCP	45.2631578947368	48.421052631578895	48.421052631578895	52.631578947368396	53.684210526315 8	45.7894736842105	46.3157894736842	49.4736842105263	48.9473684210526
BDMS	54.73684210526319	48.421052631578895	56.842105263157904	46.3157894736842	50.5263157894737	36.8421052631579	55.7894736842105	48.9473684210526	56.31578947368 42
BEC	50.0	48.421052631578895	47.8947368421053	44.7368421052632	45.7894736842105	44.2105263157895	50.0	48.9473684210526	47.8947368421053
BH	51.052631578947405	54.73684210526319	47.8947368421053	52.1052631578947	54.73684210526319	51.052631578947405	50.5263157894737	54.2105263157895	47.8947368421053
BLA	58.421052631578995	51.578947368421105	49.4736842105263	54.2105263157895	54.7368421052632 19	55.7894736842105	58.9473684210526	51.578947368421105	49.4736842105263
BTS	45.7894736842105	44.2105263157895	47.8947368421053	42.631578947368396	44.2105263157895	43.1578947368421	44.2105263157895	44.2105263157895	47.8947368421053
CBG	38.0	36.0	30.0	48.0	60.0	50.0	44.0	36.0	30.0
CENTEL	51.052631578947405	52.1052631578947	42.631578947368396	45.2631578947368	45.7894736842105	40.0	48.9473684210526	52.631578947368396	43.1578947368421
CK	49.4736842105263	44.2105263157895	43.1578947368421	56.842105263157904	47.8947368421053	41.0526315789474	50.5263157894737	44.7368421052632	43.684210526315795
CPALL	42.105263157894704	45.2631578947368	48.9473684210526	44.2105263157895	44.7368421052632	41.5789473684211	43.1578947368421	47.3684210526316	48.9473684210526
CPF	54.2105263157895	51.578947368421105	56.31578947368 42	47.8947368421053	49.4736842105263	53.6842105263158	53.6842105263158	50.5263157894737	56.31578947368 42
CPN	57.8947368421052 96	56.842105263157904	58.421052631578995	41.0526315789474	45.2631578947368	38.9473684210526	55.2631578947368	56.842105263157904	58.421052631578995
DELTA	51.052631578947405	45.7894736842105	45.7894736842105	47.3684210526316	46.842105263157904	46.842105263157904	51.578947368421105	45.7894736842105	45.7894736842105
DTAC	50.5263157894737	51.578947368421105	50.5263157894737	51.578947368421105	53.684210526315 8	51.578947368421105	52.631578947368396	51.578947368421105	51.052631578947405
EGCO	47.3684210526316	43.684210526315795	44.2105263157895	48.421052631578895	52.631578947368396	41.0526315789474	47.3684210526316	43.1578947368421	43.684210526315795
GLOW	46.3157894736842	48.9473684210526	43.684210526315795	47.3684210526316	47.8947368421053	43.1578947368421	44.7368421052632	48.9473684210526	43.684210526315795
GPSC	50.0	43.75	31.25	43.75	43.75	43.75	50.0	31.25	31.25
HMPRO	56.3157894736842	55.7894736842105	52.1052631578947	47.8947368421053	52.1052631578947	57.894736842105296	56.842105263157904	56.842105263157904	52.631578947368396
INTUCH	44.2105263157895	45.7894736842105	42.631578947368396	47.8947368421053	48.9473684210526	47.8947368421053	45.2631578947368	46.3157894736842	42.631578947368396
IRPC	46.842105263157904	50.0	45.2631578947368	43.1578947368421	45.7894736842105	50.5263157894737	46.3157894736842	50.0	45.2631578947368
IVL	51.052631578947405	55.2631578947368	48.421052631578895	47.8947368421053	43.1578947368421	54.2105263157895	52.1052631578947	54.2105263157895	48.421052631578895
KBANK	44.2105263157895	49.4736842105263	47.8947368421053	50.0	49.4736842105263	53.1578947368421	43.684210526315795	50.0	47.8947368421053
KCE	45.7894736842105	47.3684210526316	48.9473684210526	46.3157894736842	44.7368421052632	42.105263157894704	45.7894736842105	46.3157894736842	48.9473684210526
KTB	45.7894736842105	48.9473684210526	39.4736842105263	47.8947368421053	43.1578947368421	41.5789473684211	47.3684210526316	48.421052631578895	38.9473684210526
LH	46.842105263157904	43.1578947368421	41.5789473684211	40.0	44.2105263157895	43.684210526315795	46.842105263157904	42.631578947368396	41.5789473684211
MINT	48.9473684210526	50.5263157894737	42.631578947368396	44.7368421052632	45.7894736842105	40.5263157894737	48.421052631578895	49.4736842105263	43.1578947368421
MTLS	59.1836734693878	53.0612244897959	48.9795918367347	40.8163265306122	51.020408163265294	36.7346938777551	57.142857142857096	59.1836734693878	46.9387755102041
PS	50.0	50.5263157894737	40.5263157894737	46.3157894736842	41.0526315789474	43.1578947368421	57.368421052631604	50.5263157894737	40.5263157894737
PTT	56.842105263157904	56.842105263157904	60.526315789473695	41.0526315789474	44.7368421052632	43.1578947368421	55.2631578947368	58.9473684210526	61.052631578947405
PTTEP	54.2105263157895	51.578947368421105	51.578947368421105	48.9473684210526	48.421052631578895	53.1578947368421	54.2105263157895	49.4736842105263	51.578947368421105
PTTGC	50.0	52.1052631578947	51.052631578947405	45.2631578947368	45.2631578947368	53.6842105263158	50.5263157894737	53.6842105263158	51.052631578947405
ROBINS	54.73684210526319	56.842105263157904	52.1052631578947	42.105263157894704	43.684210526315795	43.684210526315795	56.842105263157904	56.842105263157904	52.1052631578947
SAWAD	43.3333333333333	51.1111111111109	45.5555555555556	53.3333333333333	51.1111111111109	51.1111111111109	43.3333333333333	51.1111111111109	46.6666666666667
SCB	49.4736842105263	54.73684210526319	54.73684210526319	52.631578947368396	54.2105263157895	57.894736842105296	52.1052631578947	55.7894736842105	54.73684210526319
SCC	45.2631578947368	46.842105263157904	47.368421052631 6	37.368421052631604	47.8947368421052 96	40.0	48.9473684210526	45.2631578947368	46.842105263157904
TASCO	44.7368421052632	49.4736842105263	41.5789473684211	45.2631578947368	46.842105263157904	46.3157894736842	44.2105263157895	47.3684210526316	41.0526315789474
TCAP	52.1052631578947	52.1052631578947	43.684210526315795	47.3684210526316	47.8947368421053	49.4736842105263	52.631578947368396	51.052631578947405	43.684210526315795
TMB	53.1578947368421	54.2105263157895	42.105263157894704	52.1052631578947	52.631578947368396	43.684210526315795	53.1578947368421	55.2631578947368	41.5789473684211
TOP	51.052631578947405	47.8947368421053	53.1578947368421	48.9473684210526	51.578947368421105	44.7368421052632	51.052631578947405	46.842105263157904	53.1578947368421
TPIPL	43.1578947368421	44.2105263157895	43.684210526315795	54.73684210526319	55.2631578947368	51.578947368421105	42.105263157894704	47.8947368421053	44.2105263157895
TRUE	50.5263157894737	52.1052631578947	50.0	52.1052631578947	52.631578947368396	47.3684210526316	46.842105263157904	53.684210526315 8	50.0
TTW	46.842105263157904	50.0	38.4210526315789	43.684210526315795	43.1578947368421	43.1578947368421	46.3157894736842	49.4736842105263	38.4210526315789
TU	46.842105263157904	50.5263157894737	44.2105263157895	47.8947368421053	48.9473684210526	44.7368421052632	43.684210526315795	49.4736842105263	44.2105263157895
WHA	48.421052631578895	52.631578947368396	43.1578947368421	44.2105263157895	46.3157894736842	48.421052631578895	45.1578947368421	43.1578947368421	43.1578947368421

Table 74 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	46.3157894736842	47.8947368421053	44.7368421052632	47.8947368421053	48.4210526315895	47.3684210526316	46.842105263157904	47.8947368421053	44.7368421052632
AOT	52.1052631578947	50.0	43.684210526315795	45.7894736842105	46.842105263157904	39.4736842105263	52.1052631578947	50.0	43.684210526315795
BA	38.8888888888889	53.703703703703695	44.4444444444444	57.4074074074074	51.8518518518505	55.5555555555556	38.8888888888889	50.0	46.296296296296305
BANPU	51.578947368421105	53.68421052631578	54.2105263157895	47.8947368421053	51.052631578947405	52.631578947368396	52.631578947368396	52.631578947368396	54.2105263157895
BBL	52.1052631578947	52.631578947368396	45.2631578947368	50.0	48.421052631578895	44.7368421052632	53.157894736842	54.2105263157895	45.7894736842105
BCP	47.36842105263158	48.9473684210526	48.421052631578895	53.68421052631578	52.631578947368396	45.7894736842105	45.2631578947368	48.9473684210526	48.9473684210526
BDMS	54.2105263157895	51.052631578947405	56.842105263157904	51.578947368421105	50.5263157894737	36.8421052631579	54.736842105263219	48.9473684210526	56.3157894736842
BEC	49.4736842105263	47.36842105263158	48.9473684210526	47.8947368421053	48.421052631578895	45.7894736842105	48.9473684210526	45.7894736842105	48.9473684210526
BH	47.36842105263158	50.0	51.578947368421105	48.421052631578895	47.36842105263158	51.578947368421105	49.4736842105263	50.0	51.052631578947405
BLA	56.3157894736842	51.578947368421105	49.4736842105263	57.368421052631604	54.2105263157895	55.7894736842105	58.421052631578995	51.578947368421105	49.4736842105263
BTS	45.2631578947368	46.3157894736842	47.36842105263158	46.842105263157904	43.684210526315795	42.631578947368396	46.3157894736842	46.3157894736842	47.36842105263158
CBG	38.0	36.0	30.0	56.000000000000001	60.0	50.0	40.0	36.0	30.0
CENTEL	48.9473684210526	52.631578947368396	43.1578947368421	43.1578947368421	45.7894736842105	40.5263157894737	48.9473684210526	52.631578947368396	43.1578947368421
CK	49.4736842105263	45.2631578947368	39.4736842105263	56.3157894736842	54.2105263157895	52.1052631578947	49.4736842105263	45.2631578947368	40.0
CPALL	41.5789473684211	47.36842105263158	48.9473684210526	45.2631578947368	46.3157894736842	41.578947368421105	42.105263157894704	46.3157894736842	48.9473684210526
CPF	51.578947368421105	50.5263157894737	56.3157894736842	51.052631578947405	52.1052631578947	53.6842105263158	45.7894736842105	52.631578947368396	56.3157894736842
CPN	57.894736842105296	57.894736842105296	58.421052631578995	40.0	45.7894736842105	38.9473684210526	57.894736842105296	56.842105263157904	58.421052631578995
DELTA	49.4736842105263	45.7894736842105	45.7894736842105	48.9473684210526	46.3157894736842	46.842105263157904	48.9473684210526	46.3157894736842	45.7894736842105
DTAC	52.631578947368396	50.5263157894737	50.5263157894737	55.7894736842105	51.578947368421105	53.6842105263158	53.157894736842	50.5263157894737	51.052631578947405
EGCO	47.36842105263158	43.1578947368421	44.2105263157895	48.421052631578895	53.157894736842	41.052631578974	46.3157894736842	42.631578947368396	43.684210526315795
GLOW	47.36842105263158	47.36842105263158	42.105263157894704	46.3157894736842	47.36842105263158	43.1578947368421	48.9473684210526	50.5263157894737	42.105263157894704
GPSC	43.75	37.5	31.25	43.75	43.75	43.75	43.75	50.0	31.25
HMPRO	56.842105263157904	55.7894736842105	51.052631578947405	47.8947368421053	52.1052631578947	56.842105263157904	55.7894736842105	55.2631578947368	51.578947368421105
INTUCH	44.2105263157895	46.842105263157904	42.631578947368396	46.842105263157904	50.0	47.8947368421053	44.2105263157895	47.36842105263158	42.631578947368396
IRPC	46.3157894736842	51.052631578947405	41.052631578974	45.7894736842105	47.36842105263158	47.36842105263158	49.4736842105263	48.9473684210526	41.5789473684211
IVL	49.4736842105263	55.7894736842105	45.2631578947368	53.6842105263158	52.1052631578947	53.6842105263158	52.1052631578947	55.7894736842105	45.2631578947368
KBANK	49.4736842105263	54.2105263157895	47.8947368421053	49.4736842105263	48.9473684210526	50.0	49.4736842105263	54.736842105263219	48.9473684210526
KCE	46.3157894736842	43.684210526315795	49.4736842105263	43.1578947368421	42.105263157894704	43.1578947368421	43.684210526315795	43.684210526315795	49.4736842105263
KTB	44.7368421052632	49.4736842105263	39.4736842105263	44.7368421052632	43.1578947368421	41.5789473684211	49.4736842105263	50.0	38.9473684210526
LH	48.421052631578895	46.842105263157904	45.7894736842105	42.631578947368396	46.3157894736842	41.5789473684211	44.7368421052632	47.8947368421053	45.7894736842105
MINT	46.842105263157904	48.9473684210526	42.105263157894704	48.9473684210526	47.8947368421053	53.1578947368421	46.3157894736842	48.421052631578895	42.105263157894704
MTLS	61.2244897959184	57.142857142857096	48.9795918367347	40.8163265306122	46.9387755102041	36.734693877551	59.1836734693878	57.142857142857096	46.9387755102041
PS	57.368421052631604	49.4736842105263	62.105263157847	54.736842105263219	53.157894736842	34.2105263157895	57.894736842105296	48.9473684210526	61.578947368421
PTT	55.2631578947368	57.894736842105296	60.526315787497369	40.5263157894737	45.2631578947368	42.631578947368396	54.736842105263219	58.421052631578995	61.052631578947405
PTTEP	54.736842105263219	54.736842105263219	54.2105263157895	45.2631578947368	46.842105263157904	51.052631578947405	54.2105263157895	54.2105263157895	54.2105263157895
PTTGC	45.2631578947368	53.684210526315795	51.052631578947405	45.7894736842105	45.2631578947368	53.6842105263158	48.421052631578895	53.684210526315795	51.052631578947405
ROBINS	54.2105263157895	56.842105263157904	52.1052631578947	43.1578947368421	44.736842105263219	43.684210526315795	54.2105263157895	56.842105263157904	52.1052631578947
SAWAD	42.222222222222	45.555555555556	46.6666666666667	55.555555555556	52.222222222222	50.0	42.222222222222	50.0	45.555555555556
SCB	52.1052631578947	54.2105263157895	54.736842105263219	52.631578947368396	54.2105263157895	57.894736842105296	51.052631578947405	54.736842105263219	54.736842105263219
SCC	51.578947368421105	46.3157894736842	46.842105263157904	38.9473684210526	40.0	40.5263157894737	49.4736842105263	44.736842105263219	47.36842105263158
TASCO	42.631578947368396	41.5789473684211	44.2105263157895	42.631578947368396	43.684210526315795	46.3157894736842	41.052631578974	41.052631578974	44.2105263157895
TCAP	50.0	51.052631578947405	43.684210526315795	47.8947368421053	47.36842105263158	49.4736842105263	50.5263157894737	52.631578947368396	43.684210526315795
TMB	48.9473684210526	53.1578947368421	45.2631578947368	53.1578947368421	51.578947368421105	43.684210526315795	48.9473684210526	53.684210526315795	44.736842105263219
TOP	53.1578947368421	47.8947368421053	53.1578947368421	50.0	52.1052631578947	44.736842105263219	52.1052631578947	48.9473684210526	53.1578947368421
TPIPL	52.631578947368396	52.1052631578947	43.1578947368421	52.631578947368396	53.157894736842	44.2105263157895	48.421052631578895	52.631578947368396	43.1578947368421
TRUE	50.0	51.052631578947405	50.5263157894737	47.36842105263158	47.8947368421053	44.2105263157895	49.4736842105263	50.5263157894737	50.5263157894737
TTW	45.2631578947368	50.0	36.8421052631579	40.5263157894737	43.684210526315795	35.7894736842105	46.842105263157904	48.9473684210526	36.8421052631579
TU	46.842105263157904	50.5263157894737	44.2105263157895	45.7894736842105	48.9473684210526	44.736842105263219	46.842105263157904	47.8947368421053	44.2105263157895
WHA	47.36842105263158	50.0	44.2105263157895	44.2105263157895	46.3157894736842	52.631578947368396	47.8947368421053	50.0	44.2105263157895

Table 75 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.105263157894704	41.578947368421	43.684210526315795	47.8947368421053	47.36842105263158	50.0	43.1578947368421	41.578947368421	43.684210526315795
AOT	47.36842105263158	42.631578947368396	38.9473684210526	50.5263157894737	50.5263157894737	51.578947368421105	49.4736842105263	43.1578947368421	38.9473684210526
BA	37.037037037037	38.8888888888889	38.8888888888889	50.0	48.148148148148	46.296296296296305	38.8888888888889	38.8888888888889	38.8888888888889
BANPU	47.8947368421053	48.9473684210526	41.578947368421	44.2105263157895	52.1052631578947	56.3157894736842	47.8947368421053	49.4736842105263	41.578947368421
BBL	49.4736842105263	53.1578947368421	44.7368421052632	54.2105263157895	48.421052631578895	44.7368421052632	51.578947368421105	53.1578947368421	44.7368421052632
BCP	46.3157894736842	43.684210526315795	38.4210526315789	47.3684210526316	50.0	52.1052631578947	46.3157894736842	38.4210526315789	38.4210526315789
BDMS	39.4736842105263	38.9473684210526	37.368421052631604	48.9473684210526	47.36842105263158	53.6842105263158	41.052631578974	38.9473684210526	37.368421052631604
BEC	47.8947368421053	47.8947368421053	42.631578947368396	48.421052631578895	53.684210526315795	52.1052631578947	46.3157894736842	47.36842105263158	43.684210526315795
BH	50.0	52.1052631578947	49.4736842105263	54.7368421052632	53.157894736842	57.368421052631604	50.5263157894737	52.1052631578947	49.4736842105263
BLA	48.9473684210526	53.6842105263158	45.7894736842105	49.4736842105263	50.5263157894737	52.1052631578947	51.052631578947405	45.7894736842105
BTS	41.052631578974	45.2631578947368	40.5263157894737	52.1052631578947	54.2105263157895	50.0	41.052631578974	46.3157894736842	40.5263157894737
CBG	30.0	30.0	32.0	56.000000000000001	62.0	54.0	30.0	36.0	32.0
CENTEL	41.5789473684211	45.2631578947368	41.5789473684211	44.7368421052632	52.631578947368396	44.2105263157895	41.578947368421	42.631578947368396	41.5789473684211
CK	38.9473684210526	38.9473684210526	38.9473684210526	45.2631578947368	46.842105263157904	47.368421052631	40.5263157894737	38.4210526315789	38.9473684210526
CPALL	43.1578947368421	50.0	43.1578947368421	54.7368421052632	44.736842105263219	49.4736842105263	43.1578947368421	45.2631578947368	43.1578947368421
CPF	44.7368421052632	43.684210526315795	44.2105263157895	50.5263157894737	47.368421052631604	48.9473684210526	45.7894736842105	43.684210526315795	43.684210526315795
CPN	46.842105263157904	48.9473684210526	41.578947368421	52.631578947368396	48.9473684210526	49.4736842105263	46.3157894736842	50.0	41.578947368421
DELTA	51.578947368421105	48.421052631578895	45.2631578947368	52.1052631578947	48.421052631578895	58.9473684210526	51.578947368421105	49.4736842105263	45.2631578947368
DTAC	51.578947368421105	53.1578947368421	50.0	42.105263157894704	53.684210526315795	53.684210526315795	52.631578947368396	55.2631578947368	50.0
EGCO	38.9473684210526	40.5263157894737	38.9473684210526	47.8947368421053	48.9473684210526	52.1052631578947	38.9473684210526	41.052631578974	38.9473684210526
GLOW	39.4736842105263	45.7894736842105	41.578947368421	56.842105263157904	48.9473684210526	55.2631578947368	40.0	40.5263157894737	41.578947368421
GPSC	43.75	43.75	43.75	37.5	43.75	50.0	37.5	43.75	43.75
HMPRO	48.9473684210526	48.9473684210526	36.8421052631579	47.3684210526316	54.736842105263219	47.3684210526316	41.578947368421	49.4736842105263	36.8421052631579
INTUCH	45.2631578947368	48.421052631578895	42.105263157894704	49.4736842105263	48.9473684210526	50.0	42.631578947368396	47.3684210526316	42.105263157894704
IRPC	45.2631578947368	45.2631578947368	43.1578947368421	47.8947368421053	46.842105263157904	46.842105263157904	43.1578947368421	44.2105263157895	43.1578947368421
IVL	52.1052631578947	51.052631578947405	42.631578947368396	45.2631578947368	50.5263157894737	52.1052631578947	53.157894736842	48.9473684210526	42.631578947368396
KBANK	46.842105263157904	48.9473684210526	45.2631578947368	43.1578947368421	44.2105263157895	47.8947368421053	47.368421052631	50.0	45.2631578947368
KCE	40.5263157894737	43.684210526315795	40.0	52.1052631578947	45.7894736842105	45.7894736842105	40.5263157894737	44.736842105263219	40.0
KTB	35.2631578947368	41.052631578974	39.4736842105263	51.578947368421105	37.368421052631604	58.9473684210526	35.7894736842105	41.052631578974	38.4210526315789
LH	38.9473684210526	41.578947368421	38.4210526315789	51.578947368421105	47.8947368421053	43.1578947368421	38.9473684210526	38.9473684210526	38.4210526315789
MINT	42.105263157894704	46.3157894736842	42.105263157894704	53.684210526358	51.578947368421105	50.5263157894737	42.105263157894704	49.4736842105263	42.105263157894704
MTLS	57.142857142857096	55.1020408163265	53.0612244897959	51.020408163265294	48.9795918367347	63.265306122449005	57.142857142857096	57.142857142857096	55.1020408163265
PS	36.3157894736842	40.5263157894737	36.3157894736842	48.4210526315895	43.684210526315795	55.7894736842105	36.3157894736842	40.0	36.3157894736842
PTT	43.1578947368421	50.5263157894737	41.052631578974	48.4210526315895	55.2631578947368	52.631578947368396	43.684210526315795	50.0	41.052631578974
PTTEP	52.631578947368396	44.736842105263219	54.2105263157895	50.0	50.0	51.578947368421105	52.631578947368396	45.7894736842105	54.2105263157895
PTTGC	42.631578947368396	44.736842105263219	46.842105263157904	51.052631578947405	45.2631578947368	45.2631578947368	43.1578947368421	44.736842105263219	46.842105263157904
ROBINS	40.5263157894737	47.8947368421053	37.894736842105296	48.421052631578895	45.2631578947368	48.421052631578895	45.7894736842105	44.736842105263219	37.894736842105296
SAWAD	41.111111111111	48.8888888888889	41.111111111111	44.4444444444444	54.444444444439	41.111111111111	42.2222222222222	44.444444444444	41.111111111111
SCB	45.7894736842105	51.578947368421105	41.052631578974	48.9473684210526	51.578947368421105	45.2631578947368	45.7894736842105	48.9473684210526	41.052631578974
SCC	42.105263157894704	46.842105263157904	40.0	47.8947368421053	47.8947368421053	45.2631578947368	42.105263157894704	42.631578947368396	40.0
TASCO	43.1578947368421	49.4736842105263	46.3157894736842	51.578947368421105	54.2105263157895	50.5263157894737	44.2105263157895	50.5263157894737	46.3157894736842
TCAP	38.9473684210526	39.4736842105263	38.4210526315789	47.8947368421053	54.2105263157895	41.578947368421	39.4736842105263	40.5263157894737	38.4210526315789
TMB	43.1578947368421	45.2631578947368	43.684210526315795	43.684210526315795	44.2105263157895	47.8947368421053	43.1578947368421	44.736842105263219	43.684210526315795
TOP	43.1578947368421	46.842105263157904	42.105263157894704	43.684210526315795	51.578947368421105	47.368421052631	44.2105263157895	49.4736842105263	41.578947368421
TPIPL	42.631578947368396	45.2631578947368	44.2105263157895	48.421052631578895	47.8947368421053	41.052631578974	42.105263157894704	46.3157894736842	44.2105263157895
TRUE	46.3157894736842	53.1578947368421	45.2631578947368	47.8947368421053	51.052631578947405	46.842105263157904	53.157894736842	45.2631578947368
TTW	34.7368421052632	38.9473684210526	34.7368421052632	46.3157894736842	37.368421052631604	51.578947368421105	34.7368421052632	39.4736842105263	34.7368421052632
TU	42.105263157894704	39.4736842105263	41.052631578974	47.8947368421053	44.2105263157895	45.7894736842105	42.105263157894704	39.4736842105263	40.0
WHA	41.052631578974	45.2631578947368	44.7368421052632	46.842105263157904	45.2631578947368	61.052631578947405	41.052631578974	44.2105263157895	44.7368421052632

PredictDirection7 : MOF (Money Flow)

Table 76 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.631578947368396	42.631578947368396	45.2631578947368	43.157894736842	43.684210526315795	44.210526315789	42.631578947368396	42.105263157894704	45.2631578947368
AOT	56.842105263157904	57.368421052631604	57.894736842105296	56.842105263157904	57.894736842105296	58.421052631578995	56.842105263157904	57.368421052631604	57.894736842105296
BA	50.0	55.55555555555556	61.111111111111111	53.703703703703695	53.703703703703695	59.2592592592593	48.1481481481481	61.111111111111111	61.111111111111111
BANPU	42.631578947368396	43.157894736842	42.105263157894704	42.105263157894704	42.631578947368396	42.105263157894704	42.631578947368396	43.157894736842	42.105263157894704
BBL	42.631578947368396	42.105263157894704	41.05263157894474	41.05263157894474	43.157894736842	42.105263157894704	42.631578947368396	40.52631578947737	41.05263157894474
BCP	54.21052631578995	50.0	55.7894736842105	53.1578947368421	51.052631578947405	53.157894736842	52.631578947368396	51.578947368421105	55.26315789473368
BDMS	57.894736842105296	58.947368421052	59.473684210263	58.421052631578995	59.473684210263	60.0	58.421052631578995	59.473684210263	59.473684210263
BEC	44.21052631578995	43.684210526315795	44.7368421052632	43.684210526315795	43.684210526315795	45.26315789473684	44.21052631578995	43.684210526315795	44.7368421052632
BH	49.473684210263	55.26315789473368	52.631578947368396	48.94736842105263	52.10526315789474	49.473684210263	50.52631578947	55.26315789473368	52.631578947368396
BLA	49.473684210263	47.36842105263 16	48.947368421052 6	46.842105263157904	47.368421052631 6	48.947368421052 6	50.0	48.421052631578995	48.94736842105263
BTS	56.315789473684 2	54.210526315789 5	56.315789473684 2	54.210526315789 5	55.263157894736 8	54.210526315789 5	56.315789473684 2	54.210526315789 5	56.315789473684 2
CBG	70.0	70.0	68.0	70.0	70.0	68.0	70.0	70.0	68.0
CENTEL	57.894736842105296	55.789473684210 5	55.263157894736 8	56.842105263157904	56.315789473684 2	53.684210526315 8	57.368421052631604	55.263157894736 8	55.263157894736 8
CK	56.315789473684 2	56.315789473684 2	55.789473684210 5	55.263157894736 8	55.263157894736 8	55.789473684210 5	55.789473684210 5	55.263157894736 8	55.789473684210 5
CPALL	54.736842105263 19	56.315789473684 2	55.263157894736 8	52.105263157894 7	55.789473684210 5	55.789473684210 5	55.789473684210 5	56.842105263157904	55.263157894736 8
CPF	45.263157894736 8	44.210526315789 5	48.947368421052 6	44.736842105263 2	44.736842105263 2	47.894736842105 3	44.736842105263 2	44.210526315789 5	48.947368421052 6
CPN	58.947368421052 6	56.315789473684 2	60.526315789473 69	55.789473684210 5	55.263157894736 8	59.473684210263	58.421052631578995	55.789473684210 5	60.526315789473 69
DELTA	53.157894736842 1	53.157894736842 1	54.210526315789 5	51.578947368421105	54.210526315789 5	55.263157894736 8	54.210526315789 5	53.157894736842 1	53.684210526315 8
DTAC	50.0	48.947368421052 6	50.0	49.473684210263	49.473684210263	50.526315789473 7	49.473684210263	49.473684210263	50.0
EGCO	57.894736842105296	53.684210526315 8	57.368421052631604	57.368421052631604	54.210526315789 5	56.315789473684 2	58.421052631578995	53.684210526315 8	57.368421052631604
GLOW	48.947368421052 6	47.368421052631 6	51.578947368421105	45.7894736842105	46.315789473684 2	51.052631578947405	49.473684210263	47.368421052631 6	51.578947368421105
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	58.947368421052 6	58.421052631578995	60.0	58.947368421052 6	58.421052631578995	60.0	58.947368421052 6	58.421052631578995	60.526315789473 69
INTUCH	41.57894736842 11	42.631578947368396	41.05263157894474	42.105263157894704	42.105263157894704	41.05263157894474	41.05263157894474	42.631578947368396	41.05263157894474
IRPC	56.315789473684 2	54.210526315789 5	53.157894736842 1	52.631578947368396	53.157894736842 1	54.736842105263 19	55.263157894736 8	53.684210526315 8	53.157894736842 1
IVL	57.368421052631604	56.842105263157904	56.842105263157904	56.315789473684 2	55.263157894736 8	57.894736842105296	57.894736842105296	57.368421052631604	56.842105263157904
KBANK	44.21052631578995	47.368421052631 6	43.157894736842 1	45.789473684210 5	45.789473684210 5	44.210526315789 5	44.736842105263 2	46.315789473684 2	43.157894736842 1
KCE	58.421052631578995	58.947368421052 6	60.0	58.421052631578995	58.947368421052 6	60.0	58.421052631578995	58.947368421052 6	60.0
KTB	53.157894736842 1	47.894736842105 3	53.157894736842 1	51.578947368421105	48.947368421052 6	52.105263157847	53.684210526315 8	46.842105263157904	53.684210526315 8
LH	47.894736842105 3	45.263157894736 8	55.789473684210 5	45.789473684210 5	46.842105263157904	50.526315789473 7	48.421052631578995	45.263157894736 8	55.789473684210 5
MINT	57.368421052631604	57.368421052631604	57.894736842105296	57.894736842105296	56.842105263157904	57.368421052631604	57.894736842105296	57.368421052631604	57.894736842105296
MTLS	48.9795918367347	53.061224489759 5	51.020408163265294	53.061224489759 5	53.061224489759 5	53.061224489759 5	48.9795918367347	51.020408163265294	51.020408163265294
PS	51.578947368421105	56.315789473684 2	62.1052631578947	48.947368421052 6	55.789473684210 5	62.105263157847	51.578947368421105	56.842105263157904	62.1052631578947
PTT	41.57894736842 11	40.0	41.05263157894474	37.894736842105296	38.947368421052 6	39.473684210263	42.105263157894704	40.0	41.05263157894474
PTTEP	53.157894736842 1	47.368421052631 6	46.842105263157904	54.210526315789 5	48.947368421052 6	46.842105263157904	47.368421052631 6	46.842105263157904	47.368421052631 6
PTTGC	50.0	50.526315789473 7	54.736842105263 19	55.263157894736 8	52.631578947368396	54.736842105263 19	53.684210526315 8	50.526315789473 7	54.736842105263 19
ROBINS	50.0	45.263157894736 8	50.526315789473 7	48.421052631578895	46.842105263157904	47.894736842105 3	49.473684210263	46.315789473684 2	50.526315789473 7
SAWAD	46.666666666667	45.555555555556	44.444444444444	45.555555555556	47.777777777778	43.333333333333	48.888888888889	46.666666666667	44.444444444444
SCB	43.684210526315795	40.526315789473 7	42.631578947368396	43.157894736842 1	40.0	42.631578947368396	44.736842105263 2	40.526315789473 7	42.631578947368396
SCC	51.052631578947405	49.473684210263	50.0	50.526315789473 7	51.578947368421105	50.0	51.052631578947405	50.526315789473 7	50.0
TASCO	53.684210526315 8	53.684210526315 8	54.210526315789 5	51.052631578947405	54.210526315789 5	54.736842105263 19	53.684210526315 8	53.684210526315 8	54.736842105263 19
TCAP	53.157894736842 1	53.157894736842 1	53.157894736842 1	53.157894736842 1	54.736842105263 19	53.684210526315 8	54.210526315789 5	52.631578947368396	53.157894736842 1
TMB	58.947368421052 6	57.368421052631604	56.842105263157904	58.421052631578995	53.684210526315 8	57.368421052631604	58.421052631578995	56.315789473684 2	56.842105263157904
TOP	60.526315789473 69	60.0	60.0	60.0	60.526315789473 69	60.0	60.526315789473 69	60.526315789473 69	60.0
TPIPL	57.368421052631604	53.684210526315 8	53.157894736842 1	58.947368421052 6	50.526315789473 7	53.684210526315 8	54.210526315789 5	54.210526315789 5	53.684210526315 8
TRUE	46.315789473684 2	47.894736842105 3	51.052631578947405	48.421052631578895	48.421052631578895	50.0	51.578947368421105	48.421052631578995	50.526315789473 7
TTW	48.947368421052 6	46.842105263157904	55.263157894736 8	44.736842105263 2	48.421052631578895	56.315789473684 2	50.0	45.789473684210 5	54.736842105263 19
TU	56.315789473684 2	53.157894736842 1	54.210526315789 5	55.789473684210 5	55.789473684210 5	54.736842105263 19	55.263157894736 8	53.157894736842 1	54.210526315789 5
WHA	54.210526315789 5	51.052631578947405	58.421052631578995	53.157894736842 1	52.105263157847	58.421052631578995	54.210526315789 5	51.578947368421105	58.421052631578995

Table 77 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	59.473684210263	54.736842105263 19	47.8947368421053	48.947368421052 6	52.631578947368396	47.368421052631 6	60.0	54.736842105263 19	47.368421052631 6
AOT	42.105263157894704	47.368421052631 6	43.684210526315795	40.0	41.578947368421 1	43.684210526315795	41.052631578974	48.421052631578895	43.684210526315795
BA	57.4074074074074	46.296296296296305	38.8888888888889	64.81481481481481	59.2592592592593	61.111111111111	57.4074074074074	48.1481481481481	38.8888888888889
BANPU	53.157894736842 1	58.947368421052 6	56.842105263157904	47.368421052631 6	50.526315789473 7	45.263157894736 8	53.157894736842 1	58.947368421052 6	56.842105263157904
BBL	47.368421052631 6	46.842105263157904	46.315789473684 2	54.736842105263 19	51.052631578947405	50.526315789473 7	51.578947368421105	46.315789473684 2	46.315789473684 2
BCP	41.578947368421 1	49.473684210263	45.263157894736 8	42.631578947368396	45.789473684210 5	38.421052631578 9	41.578947368421 1	47.894736842105 3	45.263157894736 8
BDMS	57.368421052631604	56.315789473684 2	50.526315789473 7	47.894736842105 3	52.631578947368396	40.526315789473 7	55.263157894736 8	57.368421052631604	50.526315789473 7
BEC	50.526315789473 7	52.105263157894 7	54.736842105263 19	47.894736842105 3	49.473684210263	43.157894736842 1	50.526315789473 7	51.578947368421105	55.263157894736 8
BH	47.368421052631 6	46.842105263157904	46.842105263157904	55.263157894736 8	51.578947368421105	48.421052631578895	52.105263157894 7	46.842105263157904	47.368421052631 6
BLA	47.894736842105 3	45.789473684210 5	52.631578947368396	44.736842105263 2	47.368421052631 6	54.210526315789 5	45.263157894736 8	42.631578947368396	51.578947368421105
BTS	47.368421052631 6	46.842105263157904	47.894736842105 3	43.684210526315795	46.842105263157904	44.210526315789 5	46.842105263157904	46.842105263157904	47.894736842105 3
CBG	44.0	40.0	38.0	50.0	52.0	48.0	44.0	40.0	36.0
CENTEL	50.526315789473 7	54.736842105263 19	49.473684210263	44.210526315895	48.947368421052 6	42.105263157894704	50.0	54.210526315789 5	49.473684210263
CK	45.789473684210 5	50.0	44.736842105263 2	46.842105263157904	46.842105263157904	45.789473684210 5	48.421052631578895	50.0	44.736842105263 2
CPALL	51.578947368421105	48.947368421052 6	49.473684210263	48.947368421052 6	50.0	49.473684210263	48.947368421052 6	48.947368421052 6	49.473684210263
CPF	48.947368421052 6	49.473684210263	54.736842105263 19	50.0	52.105263157847	54.736842105263 19	48.421052631578895	46.842105263157904	54.736842105263 19
CPN	61.052631578947405	58.947368421052 6	61.052631578947405	47.368421052631 6	50.526315789473 7	40.0	60.0	57.894736842105296	61.052631578947405
DELTA	47.368421052631 6	46.842105263157904	42.105263157894704	42.631578947368396	42.105263157894704	43.684210526315795	48.947368421052 6	46.315789473684 2	42.105263157894704
DTAC	51.578947368421105	49.473684210263	52.105263157847	54.210526315795	54.210526315795	54.736842105263 19	49.473684210263	52.631578947368396	52.631578947368396
EGCO	52.105263157894 7	49.473684210263	43.684210526315795	45.789473684210 5	51.578947368421105	46.842105263157904	48.421052631578895	49.473684210263	44.210526315795
GLOW	45.263157894736 8	52.631578947368396	44.210526315795	44.210526315795	46.315789473684 2	41.578947368421 1	50.526315789473 7	52.105263157894 7	44.736842105263 2
GPSC	31.25	37.5	43.75	43.75	43.75	43.75	25.0	37.5	43.75
HMPRO	45.263157894736 8	44.736842105263 2	53.684210526315 8	44.736842105263 2	44.736842105263 2	46.315789473684 2	48.421052631578895	44.736842105263 2	53.684210526315 8
INTUCH	51.052631578947405	45.789473684210 5	48.421052631578895	50.0	51.578947368421105	52.105263157847	50.526315789473 7	47.368421052631 6	48.421052631578895
IRPC	52.105263157847	47.894736842105 3	52.105263157847	47.368421052631 6	46.842105263157904	50.0	52.105263157847	48.421052631578895	51.578947368421105
IVL	52.631578947368396	52.631578947368396	44.736842105263 2	46.315789473684 2	47.894736842105 3	53.684210526315 8	51.052631578947405	52.105263157847	44.736842105263 2
KBANK	45.263157894736 8	44.210526315789 5	47.368421052631 6	43.157894736842 1	41.578947368421 1	47.368421052631 6	44.210526315789 5	44.210526315789 5	47.368421052631 6
KCE	46.842105263157904	54.210526315789 5	47.894736842105 3	42.631578947368396	48.421052631578895	40.0	46.842105263157904	56.315789473684 2	46.842105263157904
KTB	45.789473684210 5	45.789473684210 5	42.631578947368396	44.210526315789 5	44.736842105263 2	42.105263157894704	46.842105263157904	45.263157894736 8	42.631578947368396
LH	51.578947368421105	47.368421052631 6	38.947368421052 6	48.947368421052 6	46.315789473684 2	48.421052631578895	48.947368421052 6	47.368421052631 6	39.473684210263
MINT	47.894736842105 3	46.842105263157904	41.578947368421 1	45.263157894736 8	45.789473684210 5	42.105263157894704	47.894736842105 3	46.842105263157904	42.631578947368396
MTLS	44.8979591836735	46.938775510204 1	44.8979591836735	55.1020408163265	46.938775510204 1	40.816326530612 2	48.9795918367347	48.9795918367347	44.8979591836735
PS	43.684210526315795	45.263157894736 8	52.105263157847	48.421052631578895	48.421052631578895	49.473684210263	44.736842105263 2	51.578947368421105	52.631578947368396
PTT	47.894736842105 3	52.105263157847	54.736842105263 19	43.684210526315795	42.105263157894704	41.578947368421 1	47.894736842105 3	52.105263157847	54.736842105263 19
PTTEP	55.789473684210 5	51.578947368421105	54.210526315789 5	47.368421052631 6	49.473684210263	47.368421052631 6	54.736842105263 19	51.578947368421105	54.210526315789 5
PTTGC	46.842105263157904	47.368421052631 6	50.0	42.631578947368396	42.105263157894704	55.789473684210 5	48.421052631578895	47.894736842105 3	50.0
ROBINS	40.526315789473 7	42.105263157894704	52.631578947368396	40.526315789473 7	45.263157894736 8	44.736842105263 2	44.210526315789 5	42.631578947368396	52.631578947368396
SAWAD	55.555555555556	50.0	56.666666666667	43.333333333333	41.111111111111	41.111111111111	54.444444444439	53.333333333333	56.666666666667
SCB	41.57894736842 11	47.894736842105 3	50.0	54.210526315789 5	53.157894736842 1	50.526315789473 7	42.631578947368396	48.947368421052 6	50.0
SCC	52.631578947368396	53.684210526315 8	51.052631578947405	47.368421052631 6	45.789473684210 5	43.157894736842 1	51.578947368421105	52.105263157847	51.578947368421105
TASCO	46.842105263157904	47.894736842105 3	42.105263157894704	45.263157894736 8	45.789473684210 5	46.315789473684 2	47.894736842105 3	48.947368421052 6	43.157894736842 1
TCAP	43.684210526315795	41.578947368421 1	44.210526315795	47.894736842105 3	46.842105263157904	47.894736842105 3	43.684210526315795	41.578947368421 1	44.210526315795
TMB	45.789473684210 5	49.473684210263	43.157894736842 1	39.473684210263	38.947368421052 6	44.210526315895	45.263157894736 8	50.0	43.157894736842 1
TOP	47.368421052631 6	48.947368421052 6	50.0	50.526315789473 7	48.947368421052 6	44.210526315895	46.315789473684 2	49.473684210263	50.0
TPIPL	39.473684210263	44.210526315789 5	37.368421052631604	57.894736842105296	51.578947368421105	63.684210526315 8	38.947368421052 6	43.684210526315795	37.894736842105296
TRUE	52.631578947368396	53.684210526315 8	45.263157894736 8	46.315789473684 2	49.473684210263	44.736842105263 2	55.263157894736 8	53.157894736842 1	46.315789473684 2
TTW	36.842105263159	39.473684210263	41.052631578974	38.421052631578 9	38.421052631578 9	39.473684210263	40.526315789473 7	39.473684210263	41.052631578974
TU	47.894736842105 3	47.894736842105 3	46.315789473642	50.0	50.526315789473 7	46.315789473642	46.842105263157904	48.947368421052 6	45.789473684210 5
WHA	55.789473684210 5	50.0	52.631578947368396	53.157894736842 1	52.631578947368396	50.526315789473 7	55.789473684210 5	50.0	52.631578947368396

Table 78 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	58.421052631578995	54.21052631578995	50.52631578947437	50.0	55.78947368421105	50.52631578947437	58.421052631578995	54.73684210526319	50.52631578947437
AOT	40.52631578947437	43.684210526315795	47.36842105263316	40.52631578947437	38.94736842105263	40.0	37.368421052631604	43.684210526315795	47.36842105263316
BA	57.4074074074074	48.1481481481481	38.88888888888889	59.2592592592593	51.851851851851805	55.55555555555556	50.0	51.851851851851805	38.88888888888889
BANPU	51.578947368421105	56.31578947368421	57.368421052631604	47.36842105263316	50.0	45.26315789473685	52.631578947368396	56.842105263157904	57.368421052631604
BBL	52.10526315789474	48.94736842105263	42.631578947368396	50.0	54.73684210526319	44.21052631578947	50.0	47.36842105263316	42.631578947368396
BCP	42.105263157894704	49.47368421052632	45.26315789473685	41.57894736842105	46.842105263157904	38.42105263157895	42.105263157894704	48.94736842105263	45.26315789473685
BDMS	58.421052631578995	55.78947368421105	50.52631578947437	48.94736842105263	51.052631578947405	40.52631578947437	59.473684210526316	55.78947368421105	50.52631578947437
BEC	51.578947368421105	52.10526315789474	54.73684210526319	48.94736842105263	48.94736842105263	44.21052631578947	50.0	53.157894736842105	55.26315789473685
BH	48.421052631578895	50.0	46.842105263157904	47.89473684210526	47.36842105263316	47.36842105263316	49.473684210526316	49.473684210526316	47.36842105263316
BLA	47.36842105263316	45.78947368421105	52.631578947368396	45.78947368421105	46.842105263157904	54.21052631578947	45.78947368421105	45.78947368421105	51.578947368421105
BTS	47.89473684210526	46.31578947368421	50.0	45.26315789473685	45.26315789473685	44.736842105263246	48.94736842105263	47.36842105263316	50.0
CBG	38.0	38.0	38.0	50.0	52.0	48.0	44.0	42.0	36.0
CENTEL	52.10526315789474	54.73684210526319	46.31578947368421	44.21052631578947	48.94736842105263	41.57894736842105	52.10526315789474	54.73684210526319	46.31578947368421
CK	45.26315789473685	45.78947368421105	44.21052631578947	50.526315789473685	46.842105263157904	48.94736842105263	45.78947368421105	46.31578947368421	44.21052631578947
CPALL	50.526315789473685	48.94736842105263	49.473684210526316	46.842105263157904	49.473684210526316	49.473684210526316	48.421052631578895	49.473684210526316	48.421052631578895
CPF	47.36842105263316	47.36842105263316	53.157894736842105	52.10526315789474	53.157894736842105	54.736842105263246	45.78947368421105	47.36842105263316	53.157894736842105
CPN	60.0	58.94736842105263	61.052631578947405	45.78947368421105	50.0	40.0	60.0	58.94736842105263	61.052631578947405
DELTA	48.94736842105263	46.842105263157904	42.105263157894704	40.526315789473685	41.57894736842105	43.684210526315795	46.842105263157904	46.842105263157904	42.105263157894704
DTAC	50.526315789473685	48.94736842105263	51.578947368421105	52.10526315789474	51.578947368421105	56.842105263157904	48.421052631578895	48.94736842105263	51.578947368421105
EGCO	51.052631578947405	48.94736842105263	43.684210526315795	47.89473684210526	48.421052631578895	46.842105263157904	50.526315789473685	48.94736842105263	44.21052631578947
GLOW	45.78947368421105	53.157894736842105	44.21052631578947	45.26315789473685	48.421052631578895	42.631578947368396	45.78947368421105	51.578947368421105	44.21052631578947
GPSC	25.0	37.5	43.75	43.75	43.75	43.75	25.0	37.5	43.75
HMPRO	47.36842105263316	45.78947368421105	54.736842105263246	45.26315789473685	45.26315789473685	47.89473684210526	48.421052631578895	46.31578947368421	54.736842105263246
INTUCH	50.526315789473685	46.31578947368421	48.421052631578895	50.0	52.631578947368396	52.105263157894704	48.421052631578895	46.842105263157904	48.421052631578895
IRPC	51.578947368421105	48.94736842105263	51.578947368421105	47.89473684210526	46.31578947368421	47.89473684210526	49.473684210526263	49.473684210526263	51.578947368421105
IVL	53.684210526263158	54.736842105263246	52.10526315789474	49.473684210526263	47.89473684210526	52.105263157894704	56.842105263157904	56.31578947368421	51.052631578947405
KBANK	41.052631578947405	49.473684210526263	46.31578947368421	41.052631578947405	42.631578947368396	49.473684210526263	41.578947368421105	47.89473684210526	45.26315789473685
KCE	46.31578947368421	56.842105263157904	44.736842105263246	44.210526315789474	42.105263157894704	40.0	47.89473684210526	55.78947368421105	44.736842105263246
KTB	46.842105263157904	45.26315789473685	42.631578947368396	44.736842105263246	44.736842105263246	42.105263157894704	45.78947368421105	46.842105263157904	42.631578947368396
LH	53.157894736842105	51.052631578947405	58.421052631578995	48.94736842105263	47.89473684210526	44.210526315789474	49.473684210526263	50.526315789473685	58.421052631578995
MINT	44.210526315789474	45.26315789473685	45.78947368421105	47.89473684210526	50.526315789473685	44.736842105263246	42.105263157894704	46.842105263157904	44.736842105263246
MTLS	38.77551020408163	48.97959183673469	44.89795918367347	53.06122448979592	48.97959183673469	40.816326530612244	38.77551020408163	46.93877551020408	44.89795918367347
PS	44.210526315789474	48.94736842105263	53.684210526315795	38.421052631578945	44.210526315789474	47.36842105263316	43.684210526315795	48.94736842105263	53.684210526315795
PTT	47.36842105263316	51.578947368421105	55.263157894736845	43.684210526315795	42.105263157894704	42.105263157894704	44.736842105263246	52.631578947368396	55.263157894736845
PTTEP	51.052631578947405	50.526315789473685	50.526315789473685	48.94736842105263	50.0	47.36842105263316	53.684210526315795	51.052631578947405	50.526315789473685
PTTGC	47.36842105263316	47.89473684210526	50.0	42.105263157894704	41.57894736842105	55.78947368421105	46.842105263157904	47.36842105263316	50.0
ROBINS	41.052631578947405	42.105263157894704	52.631578947368396	45.26315789473685	45.26315789473685	44.736842105263246	40.526315789473685	42.105263157894704	52.631578947368396
SAWAD	54.444444444444439	52.222222222222	56.666666666667	44.444444444444	42.222222222222	41.111111111111	53.333333333333	52.222222222222	56.666666666667
SCB	42.105263157894704	47.89473684210526	50.0	55.263157894736845	52.631578947368396	50.526315789473685	42.105263157894704	47.89473684210526	50.0
SCC	51.052631578947405	52.631578947368396	50.0	45.78947368421105	44.210526315789474	42.631578947368396	52.105263157894704	52.105263157894704	49.473684210526263
TASCO	49.473684210526263	49.473684210526263	42.105263157894704	45.26315789473685	44.736842105263246	45.78947368421105	49.473684210526263	49.473684210526263	42.631578947368396
TCAP	43.684210526315795	41.578947368421105	44.210526315789474	47.89473684210526	46.842105263157904	47.89473684210526	43.684210526315795	42.105263157894704	44.210526315789474
TMB	45.78947368421105	48.421052631578895	42.105263157894704	42.631578947368396	38.947368421052632	43.157894736842105	45.78947368421105	49.473684210526263	41.578947368421105
TOP	46.842105263157904	49.473684210526263	50.0	51.052631578947405	48.421052631578895	44.210526315789474	44.736842105263246	49.473684210526263	50.0
TPIPL	42.631578947368396	40.526315789473685	40.526315789473685	53.684210526263158	51.052631578947405	53.684210526263158	41.578947368421105	44.210526315789474	40.526315789473685
TRUE	53.157894736842105	55.78947368421105	45.78947368421105	47.89473684210526	48.947368421052632	46.842105263157904	56.315789473684211	54.210526315789474	45.78947368421105
TTW	40.526315789473685	40.526315789473685	37.368421052631604	36.315789473684211	37.894736842105263	41.052631578947405	41.578947368421105	40.526315789473685	37.368421052631604
TU	46.842105263157904	47.89473684210526	46.315789473684211	49.473684210526263	51.578947368421105	46.315789473684211	47.368421052631579	48.421052631578895	46.315789473684211
WHA	54.210526315789474	51.578947368421105	53.684210526263158	52.631578947368396	51.578947368421105	55.78947368421105	56.842105263157904	51.052631578947405	53.684210526263158

Table 79 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.157894736842105	42.105263157894704	43.157894736842105	48.947368421052632	49.473684210526263	44.736842105263246	42.105263157894704	43.157894736842105	43.157894736842105
AOT	38.947368421052632	42.631578947368396	38.947368421052632	51.052631578947405	44.736842105263246	51.578947368421105	37.894736842105296	42.631578947368396	38.947368421052632
BA	35.185185185185205	33.333333333333	38.88888888888889	48.148148148148148	44.444444444444	51.851851851851805	35.185185185185205	35.185185185185205	38.88888888888889
BANPU	41.578947368421105	39.473684210526263	41.578947368421105	45.263157894736845	45.789473684210526	43.684210526315795	41.578947368421105	41.052631578947405	41.578947368421105
BBL	42.105263157894704	43.684210526315795	44.736842105263246	51.052631578947405	50.526315789473685	49.473684210526263	42.631578947368396	45.789473684210526	44.736842105263246
BCP	38.421052631578945	43.684210526315795	38.421052631578945	58.421052631578995	47.894736842105263	43.684210526315795	40.0	43.157894736842105	38.421052631578945
BDMS	45.263157894736845	44.210526315789474	37.368421052631604	51.578947368421105	44.736842105263246	51.052631578947405	46.842105263157904	44.736842105263246	37.368421052631604
BEC	46.842105263157904	44.736842105263246	45.263157894736845	54.210526315789474	46.315789473684211	45.789473684210526	44.210526315789474	44.210526315789474	45.263157894736845
BH	50.0	45.263157894736845	49.473684210526263	55.789473684210526	46.842105263157904	55.263157894736845	50.526315789473685	46.315789473684211	49.473684210526263
BLA	49.473684210526263	47.368421052631579	45.789473684210526	44.736842105263246	44.736842105263246	41.578947368421105	49.473684210526263	46.842105263157904	45.789473684210526
BTS	41.578947368421105	44.736842105263246	40.526315789473685	46.315789473684211	45.263157894736845	54.736842105263246	42.631578947368396	41.578947368421105	40.526315789473685
CBG	40.0	42.0	32.0	46.0	48.0	38.0	36.0	42.0	32.0
CENTEL	43.157894736842105	45.263157894736845	41.578947368421105	49.473684210526263	57.368421052631604	47.894736842105263	45.263157894736845	45.263157894736845	41.578947368421105
CK	38.947368421052632	41.052631578947368	38.947368421052632	52.105263157894704	46.315789473684211	47.368421052631579	40.526315789473685	43.684210526315795	38.947368421052632
CPALL	43.157894736842105	47.368421052631579	43.157894736842105	49.473684210526263	50.0	46.315789473684211	42.631578947368396	46.315789473684211	43.157894736842105
CPF	43.684210526315795	44.736842105263246	43.684210526315795	48.421052631578895	50.526315789473685	58.421052631578995	43.684210526315795	47.368421052631579	43.684210526315795
CPN	54.210526315789474	56.315789473684211	47.368421052631579	51.052631578947405	53.157894736842105	53.157894736842105	53.684210526263158	56.842105263157904	47.368421052631579
DELTA	45.263157894736845	47.894736842105263	45.263157894736845	53.157894736842105	51.578947368421105	50.0	46.842105263157904	48.947368421052632	45.263157894736845
DTAC	52.631578947368396	46.842105263157904	48.947368421052632	53.684210526263158	52.631578947368396	43.684210526315795	50.0	46.315789473684211	48.421052631578895
EGCO	41.578947368421105	41.052631578947368	38.947368421052632	54.210526315789474	47.894736842105263	41.578947368421105	41.578947368421105	42.631578947368396	38.947368421052632
GLOW	42.631578947368396	45.263157894736845	40.0	46.842105263157904	49.473684210526263	43.157894736842105	44.736842105263246	45.263157894736845	39.473684210526263
GPSC	43.75	43.75	43.75	31.25	50.0	50.0	43.75	43.75	43.75
HMPRO	36.842105263157895	38.421052631578945	36.842105263157895	51.052631578947405	46.842105263157904	45.263157894736845	37.894736842105296	36.315789473684211	36.842105263157895
INTUCH	47.368421052631579	45.263157894736845	42.105263157894704	43.684210526315795	51.578947368421105	52.105263157894704	47.368421052631579	45.789473684210526	42.105263157894704
IRPC	44.736842105263246	48.947368421052632	43.157894736842105	55.789473684210526	46.315789473684211	51.578947368421105	46.315789473684211	49.473684210526263	43.157894736842105
IVL	50.526315789473685	48.421052631578895	42.631578947368396	45.263157894736845	47.368421052631579	56.842105263157904	50.0	48.421052631578895	42.631578947368396
KBANK	43.157894736842105	40.0	45.263157894736845	47.894736842105263	46.315789473684211	53.157894736842105	43.684210526315795	50.526315789473685	45.263157894736845
KCE	47.894736842105263	54.210526315789474	40.0	56.315789473684211	51.052631578947405	45.789473684210526	43.684210526315795	52.105263157894704	40.0
KTB	47.368421052631579	43.157894736842105	39.473684210526263	46.842105263157904	49.473684210526263	46.315789473684211	43.684210526315795	43.157894736842105	39.473684210526263
LH	38.947368421052632	41.052631578947368	38.421052631578945	44.736842105263246	47.894736842105263	45.263157894736845	39.473684210526263	40.526315789473685	38.421052631578945
MINT	41.052631578947368	43.684210526315795	42.105263157894704	49.473684210526263	52.105263157894704	47.894736842105263	41.052631578947368	45.789473684210526	42.105263157894704
MTLS	55.102040816326605	51.020408163265294	48.97959183673469	42.85714285714293	53.06122448979592	51.020408163265294	46.938775510204081	46.938775510204081	48.97959183673469
PS	40.0	42.105263157894704	36.315789473684211	53.684210526263158	44.210526315789474	48.947368421052632	39.473684210526263	40.0	36.315789473684211
PTT	42.631578947368396	48.421052631578895	41.052631578947368	50.526315789473685	47.368421052631579	53.157894736842105	42.105263157894704	50.0	41.052631578947368
PTTEP	54.736842105263246	53.157894736842105	54.210526315789474	46.315789473684211	50.526315789473685	46.315789473684211	55.263157894736845	53.157894736842105	54.210526315789474
PTTGC	45.263157894736845	49.473684210526263	46.842105263157904	51.578947368421105	47.368421052631579	47.368421052631579	50.526315789473685	47.894736842105263	46.842105263157904
ROBINS	38.421052631578945	38.421052631578945	37.894736842105296	47.894736842105263	46.315789473684211	55.263157894736845	40.0	38.947368421052632	37.894736842105296
SAWAD	45.555555555556	45.555555555556	41.111111111111	46.666666666667	51.111111111109	40.0	45.555555555556	46.666666666667	41.111111111111
SCB	42.631578947368396	43.684210526315795	41.052631578947368	51.578947368421105	46.842105263157904	54.736842105263246	42.631578947368396	44.736842105263246	41.052631578947368
SCC	43.684210526315795	44.210526315789474	40.0	48.947368421052632	37.894736842105296	47.894736842105263	43.684210526315795	44.736842105263246	40.0
TASCO	47.894736842105263	51.052631578947405	46.315789473684211	46.842105263157904	50.0	43.157894736842105	48.421052631578895	50.526315789473685	46.315789473684211
TCAP	40.526315789473685	39.473684210526263	38.421052631578945	48.947368421052632	52.631578947368396	50.526315789473685	40.526315789473685	39.473684210526263	38.421052631578945
TMB	43.684210526315795	46.842105263157904	43.684210526315795	48.947368421052632	52.105263157894704	45.789473684210526	43.684210526315795	46.315789473684211	43.684210526315795
TOP	42.105263157894704	44.736842105263246	40.0	46.315789473684211	54.210526315789474	52.631578947368396	42.105263157894704	43.684210526315795	40.0
TPIPL	41.578947368421105	43.157894736842105	44.210526315789474	49.473684210526263	42.105263157894704	53.157894736842105	41.052631578947368	39.473684210526263	44.210526315789474
TRUE	48.947368421052632	49.473684210526263	45.263157894736845	50.526315789473685	51.578947368421105	46.842105263157904	50.0	50.0	45.263157894736845
TTW	34.736842105263246	37.368421052631604	34.736842105263246	42.631578947368396	46.315789473684211	56.842105263157904	35.263157894736845	36.842105263157895	34.736842105263246
TU	37.894736842105296	48.421052631578895	40.0	48.947368421052632	48.947368421052632	51.578947368421105	37.894736842105296	48.947368421052632	40.0
WHA	45.263157894736845	44.736842105263246	44.736842105263246	41.052631578947405	51.052631578947405	54.736842105263246	45.789473684210526	47.368421052631579	44.736842105263246

PredictDirection8 : OBV (On Balance Volume)

Table 80 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.21052631578895	44.73684210526316	43.15789473684211	44.73684210526316	44.21052631578895	43.15789473684211	44.21052631578895	44.21052631578895	43.684210526315795
AOT	58.421052631578995	58.421052631578995	58.421052631578995	59.4736842105263	57.368421052631604	58.947368421052	58.421052631578995	58.9473684210526	57.894736842105296
BA	38.88888888888889	38.88888888888889	44.44444444444444	44.44444444444444	44.44444444444444	42.592592592592595	35.185185185185205	38.88888888888889	44.44444444444444
BANPU	40.5263157894737	41.57894736842105	39.4736842105263	40.5263157894737	40.5263157894737	41.052631578947	40.5263157894737	41.57894736842111	39.4736842105263
BBL	43.15789473684211	43.684210526315795	43.684210526315795	43.15789473684211	44.210526315789	45.78947368421052	42.631578947368396	43.15789473684211	43.684210526315795
BCP	45.263157894737	47.89473684210526	42.631578947368396	40.0	43.684210526315795	40.5263157894737	42.105263157894704	41.57894736842111	42.631578947368396
BDMS	62.1052631578947	62.1052631578947	58.947368421052	62.1052631578947	62.1052631578947	58.947368421052	62.1052631578947	62.1052631578947	59.4736842105263
BEC	45.7894736842105	45.263157894737	46.842105263157904	46.31578947368421	45.263157894737	46.842105263157904	45.7894736842105	45.263157894737	46.842105263157904
BH	51.052631578947405	52.631578947368396	54.2105263157895	48.947368421052	52.631578947368396	54.2105263157895	51.578947368421105	51.578947368421105	54.2105263157895
BLA	51.052631578947405	50.0	49.4736842105263	50.0	48.947368421052	48.947368421052	51.052631578947405	49.4736842105263	50.0
BTS	50.5263157894737	50.0	55.789473684210	48.421052631578895	50.0	55.263157894737	50.0	48.421052631578895	55.789473684210
CBG	70.0	70.0	68.0	68.0	70.0	70.0	70.0	70.0	68.0
CENTEL	56.842105263157904	57.894736842105296	55.263157894368	57.894736842105296	58.421052631578995	55.263157894368	56.842105263157904	57.894736842105296	55.263157894368
CK	51.578947368421105	50.0	54.2105263157895	51.052631578947405	51.052631578947405	54.2105263157895	51.578947368421105	53.684210526358	54.2105263157895
CPALL	54.73684210526319	56.842105263157904	57.894736842105296	55.263157894368	55.789473684210	56.842105263157904	54.73684210526319	57.368421052631604	57.894736842105296
CPF	51.052631578947405	50.5263157894737	49.4736842105263	51.052631578947405	51.052631578947405	47.8947368421053	51.052631578947405	50.5263157894737	47.36842105263 16
CPN	54.2105263157895	54.2105263157895	49.4736842105263	54.2105263157895	53.684210526358	58.421052631578995	54.2105263157895	54.2105263157895	59.4736842105263
DELTA	52.631578947368396	53.1578947368421	47.36842105263 16	54.2105263157895	51.578947368421105	46.842105263157904	53.684210526358	51.052631578947405	47.36842105263 16
DTAC	50.0	50.0	49.4736842105263	50.0	50.0	49.4736842105263	50.0	50.0	49.4736842105263
EGCO	55.7894736842105	57.894736842105296	55.7894736842105	57.894736842105296	57.368421052631604	58.421052631578995	55.7894736842105	56.842105263157904	55.7894736842105
GLOW	52.631578947368396	50.5263157894737	54.2105263157895	50.0	52.631578947368396	54.2105263157895	50.0	50.5263157894737	54.2105263157895
GPSC	56.25	56.25	50.0	56.25	56.25	43.75	56.25	56.25	43.75
HMPRO	52.1052631578947	51.578947368421105	54.736842105263 19	51.052631578947405	49.4736842105263	51.052631578947405	52.1052631578947	52.1052631578947	55.263157894736 8
INTUCH	41.57894736842111	41.05263157894 74	42.105263157894704	41.5789473684211	41.5789473684211	42.105263157894704	42.105263157894704	42.105263157894704	42.631578947368396
IRPC	52.1052631578947	53.1578947368421	53.6842105263158	52.631578947368396	53.1578947368421	55.263157894736 8	53.6842105263158	55.263157894736 8	54.2105263157895
IVL	48.9473684210526	46.842105263157904	51.052631578947405	48.9473684210526	47.8947368421053	53.1578947368421	48.9473684210526	47.3684210526316	51.052631578947405
KBANK	46.842105263157904	45.78947368421105	44.2105263157895	46.31578947368421	46.31578947368421	44.210526315789 5	46.842105263157904	45.78947368421105	44.2105263157895
KCE	58.421052631578995	60.0	60.526315789473 69	59.4736842105263	58.421052631578995	60.526315789473 69	59.4736842105263	59.4736842105263	60.526315789473 69
KTB	45.263157894736 8	41.57894736842111	37.368421052631604	43.15789473684211	40.5263157894737	53.1578947368421	45.263157894736 8	41.57894736842111	37.368421052631604
LH	40.0	41.052631578947 4	45.263157894736 8	40.0	40.0	48.421052631578895	39.4736842105263	41.052631578947 4	45.789473684210 5
MINT	58.421052631578995	58.421052631578995	58.421052631578995	57.894736842105296	57.894736842105296	58.421052631578995	58.421052631578995	57.894736842105296	58.421052631578995
MTLS	55.1020408163265	55.1020408163265	53.061224489759 5	55.1020408163265	51.020408163265294	53.061224489759 5	53.061224489759 5	53.061224489759 5	53.061224489759 5
PS	37.368421052631604	37.368421052631604	50.5263157894737	37.894736842105296	37.368421052631604	47.8947368421053	37.368421052631604	37.368421052631604	50.0
PTT	41.57894736842111	41.57894736842111	38.947368421052 6	41.052631578947 4	41.052631578947 4	36.315789473684 2	42.105263157894704	41.052631578947 4	38.947368421052 6
PTTEP	47.36842105263 16	47.36842105263 16	46.842105263157904	48.421052631578895	47.36842105263 16	46.315789473684 2	47.36842105263 16	47.36842105263 16	46.842105263157904
PTTGC	48.421052631578895	48.421052631578995	52.631578947368396	45.7894736842105	46.842105263157904	52.631578947368396	48.421052631578895	47.894736842105 3	53.15789473684 21
ROBINS	55.7894736842105	55.263157894736 8	56.315789473684 2	56.315789473684 2	55.789473684210 5	56.842105263157904	55.7894736842105	55.7894736842105	56.315789473684 2
SAWAD	42.2222222222222	50.0	44.444444444444 4	45.555555555556	46.666666666667	43.333333333333 3	43.333333333333 3	44.444444444444 4	44.444444444444 4
SCB	40.5263157894737	41.052631578947 4	42.105263157894704	40.0	41.052631578947 4	42.631578947368396	41.052631578947 4	41.052631578947 4	42.105263157894704
SCC	48.947368421052 6	44.736842105263 2	46.315789473684 2	46.315789473684 2	44.736842105263 2	46.315789473684 2	48.947368421052 6	45.789473684210 5	46.315789473684 2
TASCO	53.6842105263158	52.631578947368396	46.315789473684 2	55.263157894736 8	52.631578947368396	44.736842105263 2	55.7894736842105	53.6842105263158	44.210526315789 5
TCAP	54.2105263157895	53.684210526315 8	54.2105263157895	55.263157894736 8	54.2105263157895	55.263157894736 8	53.157894736842 1	53.684210526315 8	54.2105263157895
TMB	52.1052631578947	50.5263157894737	55.789473684210 5	51.052631578947405	50.0	56.842105263157904	48.9473684210526	50.5263157894737	55.789473684210 5
TOP	60.0	60.526315789473 69	59.4736842105263	60.526315789473 69	60.0	60.0	60.0	60.526315789473 69	59.4736842105263
TPIPL	47.36842105263 16	45.263157894736 8	50.5263157894737	46.315789473684 2	45.263157894736 8	51.052631578947405	47.36842105263 16	45.789473684210 5	50.5263157894737
TRUE	45.789473684210 5	44.210526315789 5	46.842105263157904	46.315789473684 2	44.736842105263 2	51.052631578947405	45.789473684210 5	44.736842105263 2	46.842105263157904
TTW	43.15789473684211	40.5263157894737	45.263157894736 8	42.105263157894704	40.0	42.631578947368396	43.15789473684211	38.421052631578 9	45.263157894736 8
TU	52.1052631578947	52.631578947368396	54.2105263157895	51.052631578947405	50.0	54.2105263157895	53.157894736842 1	50.0	53.684210526315 8
WHA	48.9473684210526	46.842105263157904	55.263157894736 8	48.9473684210526	46.842105263157904	52.631578947368396	47.8947368421053	46.315789473684 2	55.789473684210 5

Table 81 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	48.421052631578895	48.9473684210526	41.0526315789474	46.3157894736842	43.15789473684211	42.105263157894704	47.8947368421053	48.421052631578895	41.0526315789474
AOT	58.421052631578995	56.315789473684 2	60.526315789473 69	43.684210526315795	43.684210526315795	53.1578947368421	57.368421052631604	56.315789473684 2	60.526315789473 69
BA	38.88888888888889	42.592592592592595	46.296296296296305	53.703703703703695	51.851851851851 05	61.111111111111 1	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	53.15789473684211	55.789473684210 5	60.526315789473 69	50.0	58.421052631578995	55.789473684210 5	51.052631578947405	53.684210526315 8	60.526315789473 69
BBL	50.0	44.736842105263 2	50.0	50.526315789473 7	52.105263157894 7	51.052631578947405	50.0	44.736842105263 2	50.0
BCP	47.36842105263 16	41.57894736842111	55.263157894736 8	45.7894736842105	43.684210526315795	44.736842105263 2	43.1578947368421	42.631578947368396	54.736842105263 19
BDMS	62.631578947368 4	61.578947368421 1	62.631578947368 4	51.578947368421105	64.736842105263 21	62.631578947368 4	62.631578947368 4	61.57894736842111	62.631578947368 4
BEC	49.473684210526 3	53.684210526315 8	49.473684210526 3	47.8947368421053	48.421052631578895	51.052631578947405	48.947368421052 6	54.2105263157895	49.473684210526 3
BH	45.7894736842105	48.947368421052 6	46.315789473684 2	49.473684210526 3	47.8947368421053	48.947368421052 6	45.7894736842105	48.947368421052 6	46.315789473684 2
BLA	46.842105263157904	47.8947368421053	47.8947368421053	52.1052631578947	52.1052631578947	54.2105263157895	45.789473684210 5	47.8947368421053	47.8947368421053
BTS	52.1052631578947	51.578947368421105	41.0526315789474	47.8947368421053	45.263157894736 8	45.789473684210 5	52.1052631578947	51.052631578947405	40.0
CBG	48.0	57.999999999999 9	70.0	64.0	66.0	44.0	42.0	64.0	70.0
CENTEL	45.263157894736 8	42.105263157894704	38.421052631578 9	44.2105263157895	41.052631578947 4	41.052631578947 4	45.263157894736 8	42.105263157894704	38.947368421052 6
CK	49.473684210526 3	49.473684210526 3	46.315789473684 2	51.052631578947405	50.526315789473 7	55.263157894736 8	48.421052631578895	51.578947368421105	47.8947368421053
CPALL	47.8947368421053	52.631578947368396	46.842105263157904	50.526315789473 7	51.052631578947405	48.421052631578895	48.421052631578895	51.578947368421105	46.842105263157904
CPF	52.631578947368396	52.1052631578947	55.789473684210 5	54.2105263157895	53.684210526358	54.2105263157895	51.578947368421105	52.1052631578947	55.789473684210 5
CPN	57.368421052631604	57.368421052631604	63.157894736842 1	54.2105263157895	46.315789473684 2	42.105263157894704	56.842105263157904	56.842105263157904	63.157894736842 1
DELTA	57.368421052631604	56.842105263157904	48.947368421052 6	53.684210526358	54.736842105263 19	47.368421052631 6	57.894736842105296	55.789473684210 5	48.947368421052 6
DTAC	47.368421052631 6	46.842105263157904	48.947368421052 6	50.526315789473 7	51.578947368421105	47.894736842105 3	47.894736842105 3	46.842105263157904	49.473684210526 3
EGCO	53.1578947368421	53.684210526358	53.157894736842 1	47.368421052631 6	50.0	46.315789473684 2	53.157894736842 1	52.631578947368396	52.631578947368396
GLOW	46.842105263157904	49.473684210526 3	44.736842105263 2	39.473684210526 3	38.947368421052 6	45.263157894736 8	46.315789473684 2	48.421052631578895	43.684210526315795
GPSC	56.25	62.5	56.25	43.75	43.75	43.75	62.5	62.5	50.0
HMPRO	50.0	53.684210526358	50.0	52.1052631578947	52.1052631578947	45.789473684210 5	51.578947368421105	53.684210526358	50.0
INTUCH	44.736842105263 2	44.736842105263 2	44.210526315789 5	44.736842105263 2	43.684210526315795	45.263157894736 8	45.789473684210 5	44.736842105263 2	44.210526315789 5
IRPC	55.789473684210 5	50.0	51.052631578947405	51.052631578947405	52.1052631578947	51.052631578947405	55.263157894736 8	49.473684210526 3	51.052631578947405
IVL	61.57894736842111	57.894736842105296	37.368421052631604	56.842105263157904	52.105263157894 7	58.421052631578995	46.842105263157904	61.052631578947405	37.368421052631604
KBANK	50.0	46.315789473684 2	45.263157894736 8	42.105263157894704	42.105263157894704	46.842105263157904	50.0	43.15789473684211	45.263157894736 8
KCE	57.368421052631604	56.842105263157904	59.473684210526 3	51.578947368421105	54.210526315789 5	39.473684210526 3	59.473684210526 3	57.894736842105296	59.473684210526 3
KTB	54.2105263157895	54.736842105263 19	43.684210526315 8	53.684210526358	53.157894736842 1	43.684210526315795	54.2105263157895	57.368421052631604	44.210526315789 5
LH	56.842105263157904	53.1578947368421	57.368421052631604	47.8947368421053	46.842105263157904	56.315789473684 2	57.368421052631604	55.263157894736 8	57.368421052631604
MINT	57.368421052631604	57.894736842105296	57.894736842105296	58.421052631578995	58.421052631578995	57.894736842105296	57.368421052631604	57.894736842105296	57.894736842105296
MTLS	59.1836734693878	51.020408163265294	51.020408163265294	48.9795918367347	46.9387755102041	51.020408163265294	55.1020408163265	51.020408163265294	51.020408163265294
PS	42.631578947368396	43.157894736842 1	60.0	47.8947368421053	44.210526315789 5	52.105263157894 7	45.263157894736 8	42.631578947368396	60.0
PTT	45.263157894736 8	47.8947368421053	58.421052631578995	44.736842105263 2	45.263157894736 8	41.578947368421 1	43.684210526315795	47.894736842105 3	58.421052631578995
PTTEP	47.36842105263 16	49.473684210526 3	53.684210526358	50.0	46.842105263157904	50.526315789473 7	47.36842105263 16	47.8947368421053	53.684210526358
PTTGC	52.631578947368396	48.421052631578895	49.473684210526 3	58.421052631578995	53.157894736842 1	52.631578947368396	53.157894736842 1	48.421052631578895	49.473684210526 3
ROBINS	60.0	58.421052631578995	58.421052631578995	54.736842105263 19	57.368421052631604	58.947368421052 6	60.0	58.421052631578995	58.421052631578995
SAWAD	55.555555555556	54.444444444444 39	61.111111111111 1	42.2222222222222	44.444444444444 4	43.333333333333 3	53.333333333333 3	54.444444444444 39	61.111111111111 1
SCB	51.052631578947405	51.052631578947405	48.421052631578895	50.0	47.8947368421053	48.947368421052 6	47.368421052631 6	50.0	47.8947368421053
SCC	51.052631578947405	51.052631578947405	55.263157894736 8	53.157894736842 1	53.684210526358	56.315789473684 2	48.947368421052 6	48.947368421052 6	55.263157894736 8
TASCO	47.8947368421053	46.842105263157904	46.315789473684 2	47.8947368421053	46.315789473684 2	46.315789473684 2	47.8947368421053	47.368421052631 6	46.315789473684 2
TCAP	53.6842105263158	55.789473684210 5	56.842105263157904	48.947368421052 6	52.105263157894 7	58.421052631578995	55.789473684210 5	56.315789473684 2	56.842105263157904
TMB	50.5263157894737	51.052631578947405	46.842105263157904	40.0	41.578947368421 1	43.157894736842 1	51.578947368421105	47.8947368421053	46.842105263157904
TOP	50.0	48.421052631578895	53.157894736842 1	43.684210526315795	41.578947368421 1	44.210526315789 5	53.157894736842 1	49.473684210526 3	53.157894736842 1
TPIPL	56.3157894736842	54.2105263157895	51.578947368421105	57.894736842105296	60.0	54.210526315789 5	56.842105263157904	55.789473684210 5	51.578947368421105
TRUE	45.263157894736 8	47.8947368421053	50.0	48.947368421052 6	48.947368421052 6	48.947368421052 6	48.421052631578895	54.2105263157895	46.842105263157904
TTW	49.473684210526 3	66.315789473684 19	52.1052631578947	50.526315789473 7	50.0	34.736842105263 2	51.052631578947405	66.315789473684 19	52.631578947368396
TU	41.052631578947 4	48.421052631578895	46.842105263157904	50.526315789473 7	48.947368421052 6	47.368421052631 6	44.210526315789 5	49.473684210526 3	46.842105263157904
WHA	54.736842105263 19	50.526315789473 7	52.631578947368396	48.947368421052 6	49.473684210526 3	44.736842105263 2	51.578947368421105	51.052631578947405	52.631578947368396

Table 82 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.7368421052632	42.6315789473896	45.2631578947368	46.3157894736842	43.15789473684421	43.684210526315795	47.3684210526316	43.15789473684421	45.26315789473684
AOT	57.894736842105296	56.842105263157904	60.526315789473369	46.31578947368	53.68421052631158	56.842105263157904	57.894736842105296	57.368421052631604	60.526315789473369
BA	37.037037037037	35.185185185185205	44.444444444444	61.111111111111111	57.407407074074	64.814814814814814	35.185185185185205	35.185185185185205	44.444444444444
BANPU	52.1052631578947	55.7894736842105	60.0	53.15789473684421	57.368421052631604	54.210526315895	52.631578947368396	55.7894736842105	60.0
BBL	48.421052631578895	42.631578947368396	50.0	47.36842105263316	51.578947368421105	51.578947368421105	48.9473684210526	44.2105263157895	48.9473684210526
BCP	47.36842105263316	42.105263157894704	55.263157894736	45.7894736842105	43.684210526315795	44.7368421052632	46.3157894736842	42.105263157894704	54.736842105263319
BDMS	62.6315789473684	61.57894736842211	62.6315789473684	52.1052631578947	65.263157894736	62.63157894736	62.63157894736	61.57894736842211	62.6315789473684
BEC	48.9473684210526	52.1052631578947	52.1052631578947	47.36842105263316	50.0	51.052631578947405	49.4736842105263	52.1052631578947	52.1052631578947
BH	51.052631578947405	50.0	47.8947368421053	53.68421052631158	51.578947368421105	48.421052631578895	48.9473684210526	48.9473684210526	47.8947368421053
BLA	47.36842105263316	47.8947368421053	47.8947368421053	52.631578947368396	52.1052631578947	54.210526315895	46.842105263157904	45.263157894736	47.8947368421053
BTS	52.1052631578947	53.1578947368421	41.0526315789474	43.1578947368421	45.263157894736	41.0526315789474	50.0	53.1578947368421	40.52631578947737
CBG	42.0	64.0	70.0	64.0	60.0	44.0	46.0	64.0	70.0
CENTEL	43.684210526315795	41.57894736842211	38.4210526315789	44.7368421052632	40.52631578947737	43.684210526315795	43.684210526315795	41.0526315789474	39.47368421052263
CK	47.8947368421053	43.1578947368421	46.842105263157904	52.631578947368396	53.15789473684421	56.3157894736842	48.421052631578895	43.684210526315795	45.789473684210
CPALL	47.36842105263316	51.052631578947405	46.842105263157904	51.052631578947405	51.578947368421105	48.421052631578895	47.3684210526316	52.1052631578947	46.842105263157904
CPF	53.68421052631158	52.631578947368396	55.7894736842105	53.68421052631158	57.368421052631604	54.210526315895	50.5263157894737	53.1578947368421	55.7894736842105
CPN	57.894736842105296	57.368421052631604	63.157894736821	50.0	45.789473684210	42.105263157894704	56.842105263157904	57.368421052631604	63.157894736821
DELTA	57.368421052631604	55.2631578947368	48.9473684210526	55.2631578947368	55.789473684210	47.368421052631	58.421052631578995	57.368421052631604	48.9473684210526
DTAC	50.5263157894737	46.3157894736842	50.5263157894737	53.1578947368421	56.842105263157904	47.8947368421053	50.0	45.789473684210	50.5263157894737
EGCO	53.1578947368421	54.2105263157895	53.1578947368421	50.5263157894737	49.4736842105263	46.3157894736842	53.68421052631158	54.2105263157895	52.631578947368396
GLOW	47.8947368421053	45.789473684210	44.7368421052632	40.5263157894737	38.421052631579	45.263157894736	46.842105263157904	47.8947368421053	45.263157894736
GPSC	56.25	56.25	56.25	43.75	43.75	43.75	62.5	62.5	50.0
HMPRO	49.4736842105263	54.2105263157895	49.4736842105263	53.1578947368421	51.578947368421105	45.789473684210	50.0	54.2105263157895	49.4736842105263
INTUCH	45.789473684210	45.789473684210	44.2105263157895	47.36842105263316	43.684210526315795	45.263157894736	44.7368421052632	45.263157894736	44.2105263157895
IRPC	55.263157894736	50.5263157894737	51.578947368421105	50.5263157894737	51.578947368421105	50.0	55.263157894736	51.052631578947405	51.578947368421105
IVL	59.4736842105263	60.526315789473369	42.105263157894704	55.263157894736	54.736842105263319	51.578947368421105	57.368421052631604	58.421052631578995	41.57894736842211
KBANK	46.3157894736842	45.789473684210	45.263157894736	41.5789473684211	42.105263157894704	44.210526315895	48.421052631578895	45.263157894736	45.263157894736
KCE	60.0	57.368421052631604	59.4736842105263	50.5263157894737	51.052631578947405	37.368421052631604	59.4736842105263	57.368421052631604	58.9473684210526
KTB	53.157894736821	55.789473684210	43.684210526315795	53.68421052631158	54.736842105263319	43.684210526315795	55.789473684210	57.368421052631604	44.210526315895
LH	55.789473684210	52.631578947368396	59.4736842105263	57.368421052631604	54.736842105263319	51.052631578947405	57.368421052631604	53.1578947368421	59.4736842105263
MINT	60.526315789473369	58.421052631578995	57.894736842105296	62.631578947684	63.157894736821	57.894736842105296	61.052631578947405	58.421052631578995	57.894736842105296
MTLS	53.0612244897959	53.0612244897959	51.020408163265294	42.8571428571429	42.8571428571429	51.020408163265294	55.102040816326	51.020408163265294	51.020408163265294
PS	46.315789473684	45.789473684210	61.052631578947405	52.631578947368396	53.684210526316	62.631578947368	47.368421052631	45.789473684210	61.052631578947405
PTT	45.263157894736	47.8947368421053	58.421052631578995	45.263157894736	46.315789473684	41.578947368421	43.157894736842	47.368421052631	58.421052631578995
PTTEP	48.9473684210526	47.8947368421053	56.842105263157904	45.263157894736	45.263157894736	46.842105263157904	47.8947368421053	47.368421052631	56.842105263157904
PTTGC	52.1052631578947	48.421052631578895	49.4736842105263	55.789473684210	55.263157894736	52.631578947368396	51.578947368421105	47.8947368421053	49.4736842105263
ROBINS	60.0	58.421052631578995	58.421052631578995	54.736842105263319	56.842105263157904	58.9473684210526	60.526315789473369	58.421052631578995	58.421052631578995
SAWAD	55.555555555556	53.333333333333	61.111111111111111	48.88888888888889	40.0	44.444444444444	57.777777777779	53.333333333333	61.111111111111111
SCB	47.8947368421053	51.052631578947405	48.421052631578895	50.0	47.8947368421053	48.9473684210526	50.0	50.5263157894737	47.8947368421053
SCC	48.9473684210526	50.5263157894737	54.736842105263319	52.631578947368396	54.736842105263319	55.789473684210	50.0	51.052631578947405	54.736842105263319
TASCO	48.9473684210526	48.9473684210526	46.3157894736842	45.789473684210	45.789473684210	44.210526315895	49.4736842105263	49.4736842105263	45.789473684210
TCAP	54.2105263157895	57.894736842105296	56.842105263157904	50.0	52.1052631578947	58.421052631578995	54.2105263157895	56.3157894736842	56.842105263157904
TMB	51.052631578947405	47.8947368421053	46.3157894736842	40.0	41.052631578947	43.684210526315795	50.5263157894737	48.421052631578895	46.3157894736842
TOP	51.578947368421105	46.3157894736842	53.157894736821	43.684210526315795	43.157894736821	44.210526315895	53.684210526315	51.052631578947405	53.157894736821
TPIPL	56.3157894736842	54.2105263157895	46.842105263157904	57.894736842105296	55.789473684210	51.052631578947405	58.9473684210526	54.2105263157895	46.3157894736842
TRUE	50.0	50.5263157894737	48.421052631578895	47.8947368421053	48.947368421052	47.368421052616	49.4736842105263	49.4736842105263	48.421052631578895
TTW	54.736842105263319	61.57894736842211	45.2631578947368	49.4736842105263	50.0	49.4736842105263	53.684210526315	61.57894736842211	45.2631578947368
TU	41.5789473684211	52.631578947368396	46.842105263157904	50.0	48.947368421026	47.368421052616	41.5789473684211	51.052631578947405	46.842105263157904
WHA	54.736842105263319	50.5263157894737	49.4736842105263	49.4736842105263	48.947368421026	43.684210526315795	52.1052631578947	51.052631578947405	49.4736842105263

Table 83 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.631578947368396	42.631578947368396	43.684210526315795	52.631578947368396	42.631578947368396	46.842105263157904	40.5263157894737	41.57894736842211	43.684210526315795
AOT	48.9473684210526	51.052631578947405	58.421052631578995	58.421052631578995	57.368421052631604	43.157894736821	55.263157894736	50.526315789473	58.421052631578995
BA	46.296296296296305	44.444444444444	38.88888888888889	50.0	42.592592592592595	38.88888888888889	46.296296296296305	44.444444444444	38.88888888888889
BANPU	52.1052631578947	48.9473684210526	41.57894736842211	50.0	48.9473684210526	54.736842105263319	50.0	48.9473684210526	41.57894736842211
BBL	47.8947368421053	49.4736842105263	44.7368421052632	50.5263157894737	55.263157894736	48.421052631578895	47.8947368421053	50.0	44.736842105263
BCP	42.631578947368396	43.157894736821	38.4210526315789	53.1578947368421	50.5263157894737	55.7894736842105	43.157894736821	45.789473684210	38.4210526315789
BDMS	52.1052631578947	52.631578947368396	62.1052631578947	47.8947368421053	57.368421052631604	46.3157894736842	52.1052631578947	57.368421052631604	62.1052631578947
BEC	46.842105263157904	46.842105263157904	45.2631578947368	52.1052631578947	45.263157894736	51.578947368421105	46.842105263157904	47.8947368421053	44.736842105263
BH	48.9473684210526	51.578947368421105	49.4736842105263	48.9473684210526	41.578947368421	50.0	53.684210526358	51.052631578947405	49.4736842105263
BLA	48.9473684210526	48.9473684210526	45.789473684210	42.1052631578704	53.157894736421	51.578947368421105	49.4736842105263	50.0	45.789473684210
BTS	41.578947368421	41.0526315789474	40.5263157894737	46.3157894736842	45.789473684210	58.421052631578995	40.5263157894737	39.4736842105263	40.52631578947737
CBG	46.0	46.0	32.0	30.0	46.0	56.000000000001	46.0	44.0	32.0
CENTEL	41.0526315789474	43.1578947368421	41.5789473684211	46.842105263157904	48.9473684210526	39.473684210263	41.0526315789474	43.1578947368421	41.57894736842211
CK	38.9473684210526	39.4736842105263	38.9473684210526	43.1578947368421	40.0	42.631578947368396	38.9473684210526	38.9473684210526	38.9473684210526
CPALL	43.1578947368421	43.1578947368421	43.1578947368421	46.842105263157904	45.789473684210	45.263157894736	43.1578947368421	45.789473684210	43.1578947368421
CPF	44.21052631578955	51.578947368421105	43.684210526315795	57.368421052631604	54.210526315895	46.842105263157904	44.7368421052632	48.9473684210526	43.684210526315795
CPN	53.684210526358	50.5263157894737	38.4210526315789	53.684210526358	62.1052631578947	48.421052631578895	53.157894736821	51.578947368421105	38.4210526315789
DELTA	50.0	51.052631578947405	45.789473684210	54.210526315895	55.789473684210	51.052631578947405	50.5263157894737	51.578947368421105	45.789473684210
DTAC	44.21052631578955	43.684210526315795	48.947368421026	53.1578947368421	45.263157894736	48.421052631578895	44.7368421052632	43.684210526315795	49.4736842105263
EGCO	44.21052631578955	37.894736842105296	42.631578947368396	45.263157894736	43.684210526315795	43.684210526315795	42.631578947368396	40.0	42.631578947368396
GLOW	44.21052631578955	44.736842105263	41.0526315789474	55.263157894736	48.947368421026	52.631578947368396	44.736842105263	44.736842105263	41.0526315789474
GPSC	56.25	43.75	43.75	43.75	43.75	50.0	43.75	50.0	43.75
HMPRO	45.789473684210	45.263157894736	36.842105263179	59.4736842105263	51.578947368421105	42.105263157894704	44.210526315895	38.421052631579	36.842105263179
INTUCH	44.736842105263	44.736842105263	44.210526315895	45.789473684210	48.421052631578895	55.263157894736	44.736842105263	46.3157894736842	44.210526315895
IRPC	52.631578947368396	55.789473684210	43.157894736821	50.0	46.842105263157904	55.263157894736	53.684210526358	55.263157894736	43.157894736821
IVL	52.631578947368396	45.789473684210	42.631578947368396	56.3157894736842	49.4736842105263	45.789473684210	51.578947368421105	43.157894736821	42.631578947368396
KBANK	45.263157894736	45.789473684210	45.263157894736	48.947368421026	46.842105263157904	44.736842105263	45.263157894736	45.789473684210	45.263157894736
KCE	49.4736842105263	55.263157894736	60.0	50.5263157894737	47.368421052616	40.0	51.052631578947405	51.052631578947405	60.0
KTB	40.0	43.157894736821	39.4736842105263	44.736842105263	60.526315789473369	47.894736842105	40.0	48.421052631578895	39.473684210263
LH	42.631578947368396	50.0	38.4210526315789	40.0	53.157894736821	41.57894736842211	43.157894736821	46.842105263157904	38.4210526315789
MINT	54.210526315895	54.736842105263319	41.0526315789474	54.736842105263319	61.052631578947405	42.105263157894704	54.736842105263319	48.947368421026	40.526315789473
MTLS	51.020408163265294	46.9387755102041	51.020408163265294	46.9387755102041	55.102040816365	63.265306122449005	51.020408163265294	46.9387755102041	51.020408163265294
PS	33.684210526358	37.894736842105296	36.315789473642	38.947368421026	48.947368421026	49.473684210263	33.684210526358	38.421052631789	36.315789473642
PTT	52.631578947368396	53.684210526358	52.1052631578947	55.263157894736	52.631578947368396	50.0	53.157894736821	54.210526315895	52.1052631578947
PTTEP	53.684210526358	52.631578947368396	54.210526315895	51.052631578947405	50.5263157894737	46.3157894736842	53.684210526358	52.1052631578947	54.210526315895
PTTGC	55.263157894736	58.421052631578995	48.9473684210526	50.5263157894737	48.947368421026	46.842105263157904	48.421052631578895	56.842105263157904	48.947368421026
ROBINS	56.842105263157904	56.842105263157904	57.894736842105296	54.210526315895	61.052631578947405	39.473684210263	57.368421052631604	61.57894736842211	55.263157894736
SAWAD	41.111111111111	41.111111111111	41.111111111111	48.88888888888889	40.0	43.333333333333	41.111111111111	42.222222222222	41.111111111111
SCB	49.4736842105263	46.3157894736842	41.0526315789474	46.842105263157904	51.052631578947405	46.3157894736842	50.0	47.368421052616	41.0526315789474
SCC	48.9473684210526	55.263157894736	40.0	52.631578947368396	48.421052631578895	52.105263157894	49.4736842105263	53.684210526358	40.0
TASCO	46.3157894736842	46.842105263157904	46.3157894736842	50.5263157894737	47.894736842105	55.789473684210	46.3157894736842	47.368421052616	46.3157894736842
TCAP	42.105263157894704	42.631578947368396	38.4210526315789	49.4736842105263	51.052631578947405	38.421052631579	41.5789473684211	43.157894736821	38.4210526315789
TMB	43.684210526315795	43.157894736821	43.684210526315795	51.578947368421105	48.421052631578895	52.105263157894	43.684210526315795	42.105263157894704	43.684210526315795
TOP	47.368421052616	43.684210526315795	47.894736842105	45.789473684210	41.578947368421	47.894736842105	48.421052631578895	45.789473684210	47.894736842105
TPIPL	40.526315789473	47.894736842105	40.0	53.684210526358	46.842105263157904	44.210526315895	41.578947368421	48.947368421026	40.0
TRUE	49.4736842105263	48.421052631578895	45.263157894736	53.157894736421	55.789473684210	54.736842105263319	49.4736842105263	48.421052631578895	45.263157894736
TTW	34.736842105263	41.57894736842211	34.736842105263	54.210526315895	46.842105263157904	57.368421052631604	43.684210526315795	42.105263157894704	34.736842105263
TU	38.9473684210526	46.842105263157904	40.0	49.4736842105263	52.105263157894	44.736842105263	38.421052631579	46.315789473642	40.0
WHA	44.736842105263	52.631578947368396	44.736842105263	45.789473684210	54.210526315895	40.526315789473	46.315789473642	48.947368421026	44.736842105263

PredictDirection9 : RSI (Relative Strength Index)

Table 84 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.7368421052632	42.6315789474396	44.2105263157895	44.2105263157895	42.6315789474396	43.157894736842	44.2105263157895	43.157894736842	44.2105263157895
AOT	56.3157894736842	56.3157894736842	57.8947368421052	56.3157894736842	56.3157894736842	57.8947368421052	55.7894736842105	56.8421052631579	57.8947368421052
BA	46.2962962962963	51.8518518518518	46.2962962962963	51.8518518518518	55.5555555555556	55.5555555555556	46.2962962962963	50.0	46.2962962962963
BANPU	39.4736842105263	38.9473684210526	40.5263157894737	40.0	38.9473684210526	40.5263157894737	39.4736842105263	38.9473684210526	40.5263157894737
BBL	44.2105263157895	42.1052631578947	42.1052631578947	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	42.1052631578947
BCP	52.6315789473396	54.2105263157895	54.7368421052639	53.1578947368421	52.6315789473396	55.2631578947368	53.6842105263158	54.7368421052639	54.7368421052639
BDMS	56.8421052631579	56.8421052631579	56.3157894736842	56.8421052631579	56.3157894736842	56.8421052631579	56.8421052631579	56.8421052631579	56.3157894736842
BEC	44.7368421052632	45.2631578947368	45.7894736842105	44.7368421052632	44.7368421052632	45.2631578947368	44.7368421052632	45.2631578947368	45.7894736842105
BH	53.1578947368421	52.6315789473396	54.2105263157895	53.1578947368421	52.6315789473396	54.2105263157895	53.1578947368421	53.1578947368421	54.2105263157895
BLA	51.0526315789474	48.4210526315789	48.4210526315789	50.5263157894737	48.4210526315789	47.8947368421053	51.0526315789474	48.4210526315789	48.4210526315789
BTS	56.3157894736842	53.6842105263158	57.3684210526316	55.7894736842105	53.1578947368421	54.2105263157895	56.3157894736842	53.6842105263158	58.4210526315789
CBG	70.0	68.0	68.0	70.0	68.0	68.0	70.0	68.0	68.0
CENTEL	55.2631578947368	54.7368421052639	56.3157894736842	55.2631578947368	54.7368421052639	56.3157894736842	55.2631578947368	55.7894736842105	56.3157894736842
CK	53.1578947368421	54.2105263157895	55.7894736842105	52.6315789473396	53.1578947368421	55.2631578947368	56.8421052631579	54.7368421052639	55.7894736842105
CPALL	53.6842105263158	54.7368421052639	54.7368421052639	53.6842105263158	53.6842105263158	52.6315789473396	52.6315789473396	55.2631578947368	54.7368421052639
CPF	50.5263157894737	50.5263157894737	47.8947368421053	51.0526315789474	50.0	47.3684210526316	50.5263157894737	50.0	48.4210526315789
CPN	56.3157894736842	57.8947368421052	58.9473684210526	55.7894736842105	57.8947368421052	55.7894736842105	55.7894736842105	57.8947368421052	56.0
DELTA	58.9473684210526	55.7894736842105	55.2631578947368	56.8421052631579	56.8421052631579	55.2631578947368	57.3684210526316	56.3157894736842	55.2631578947368
DTAC	50.5263157894737	50.0	50.5263157894737	50.5263157894737	49.4736842105263	50.0	50.5263157894737	49.4736842105263	50.5263157894737
EGCO	55.7894736842105	54.2105263157895	55.7894736842105	56.8421052631579	56.8421052631579	55.2631578947368	55.7894736842105	54.7368421052639	55.7894736842105
GLOW	47.3684210526316	45.7894736842105	52.6315789473396	46.3157894736842	46.8421052631579	51.0526315789405	46.8421052631579	44.7368421052632	52.1052631578947
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	60.5263157894369	60.5263157894369	57.8947368421052	57.3684210526316	60.0	58.9473684210526	60.5263157894369	61.0526315789405	57.8947368421052
INTUCH	42.1052631578947	41.0526315789474	42.6315789473396	41.0526315789474	41.5789473684211	42.1052631578947	41.5789473684211	41.0526315789474	42.6315789473396
IRPC	53.1578947368421	53.6842105263158	53.1578947368421	52.6315789473396	54.2105263157895	54.2105263157895	54.2105263157895	54.7368421052639	53.1578947368421
IVL	57.3684210526316	56.3157894736842	55.7894736842105	56.8421052631579	55.2631578947368	54.7368421052639	57.8947368421052	54.7368421052639	55.7894736842105
KBANK	47.3684210526316	45.2631578947368	45.2631578947368	44.2105263157895	46.3157894736842	44.2105263157895	45.7894736842105	45.2631578947368	45.2631578947368
KCE	59.4736842105263	60.0	60.0	59.4736842105263	60.0	60.0	60.0	60.0	60.0
KTB	47.3684210526316	45.2631578947368	51.5789473684211 05	47.3684210526316	48.4210526315789 5	48.9473684210526	46.8421052631579 4	46.8421052631579 4	51.0526315789474 05
LH	49.4736842105263	48.9473684210526	56.8421052631579 4	50.5263157894737	51.0526315789474 05	54.2105263157895	48.9473684210526	48.9473684210526	56.8421052631579 4
MINT	58.4210526315789 5	56.8421052631579 4	57.8947368421052 96	57.8947368421052 96	57.3684210526316 04	57.3684210526316 04	58.4210526315789 5	56.8421052631579 4	57.8947368421052 96
MTLS	51.0204081632652 94	53.0612244897959	53.0612244897959	48.9795918367347	51.0204081632652 94	55.1020408163265	51.0204081632652 94	55.1020408163265	53.0612244897959
PS	54.2105263157895	55.2631578947368	64.2105263157895	48.9473684210526	54.7368421052639	65.2631578947368	54.7368421052639	55.7894736842105	64.2105263157895
PTT	44.2105263157895	40.5263157894737	41.5789473684211	38.9473684210526	39.4736842105263	38.9473684210526	43.6842105263157 95	42.1052631578947 04	41.5789473684211
PTTEP	49.4736842105263	45.7894736842105	45.7894736842105	48.4210526315789 5	45.2631578947368	45.7894736842105	49.4736842105263	46.3157894736842	45.7894736842105
PTTGC	51.5789473684211 05	53.1578947368421	53.6842105263158	54.7368421052639	52.6315789473396	53.6842105263158	52.6315789473396	52.6315789473396	53.6842105263158
ROBINS	54.2105263157895	52.1052631578947	50.5263157894737	52.1052631578947	52.1052631578947	52.6315789473396	54.2105263157895	53.1578947368421	50.5263157894737
SAWAD	51.1111111111109	50.0	46.6666666666667	50.0	47.7777777777778	51.1111111111109	51.1111111111109	53.3333333333333	47.7777777777778
SCB	41.0526315789474	40.5263157894737	41.0526315789474	41.0526315789474	40.0	40.5263157894737	41.0526315789474	40.0	41.0526315789474
SCC	47.8947368421053	45.2631578947368	50.5263157894737	49.4736842105263	46.8421052631579 4	47.3684210526316	48.4210526315789 5	45.2631578947368	50.5263157894737
TASCO	54.2105263157895	52.6315789473396	57.3684210526316 04	56.3157894736842	52.1052631578947	57.3684210526316 04	53.6842105263158	53.1578947368421	57.3684210526316 04
TCAP	52.6315789473396	54.7368421052639	53.6842105263158	54.7368421052639	54.7368421052639	54.2105263157895	51.5789473684211 05	53.6842105263158	53.6842105263158
TMB	56.8421052631579 4	57.3684210526316 04	58.9473684210526	60.5263157894736 9	54.7368421052639	59.4736842105263	57.8947368421052 96	57.3684210526316 04	58.9473684210526
TOP	60.5263157894736 9	60.0	60.0	61.0526315789474 05	60.0	60.0	60.5263157894736 9	60.0	60.0
TPIPL	56.8421052631579 4	53.1578947368421	54.7368421052639	55.2631578947368	53.1578947368421	56.8421052631579 4	57.3684210526316 04	53.1578947368421	54.7368421052639
TRUE	44.2105263157895	45.7894736842105	55.7894736842105	46.8421052631579 4	47.3684210526316	54.7368421052639	43.6842105263157 95	48.9473684210526	55.2631578947368
TTW	45.7894736842105	51.0526315789474 05	60.0	44.7368421052632	42.6315789473396	62.1052631578947	46.3157894736842	42.1052631578947 04	60.0
TU	51.0526315789474 05	50.5263157894737	57.3684210526316 04	51.0526315789474 05	52.1052631578947	57.3684210526316 04	53.1578947368421	52.6315789473396	57.3684210526316 04
WHA	51.5789473684211 05	54.2105263157895	58.4210526315899 5	52.6315789473396	55.2631578947368	57.8947368421052 96	54.2105263157895	53.6842105263158	58.9473684210526

Table 85 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	56.3157894736842	52.1052631578947	45.2631578947368	51.5789473684211 05	54.7368421052639	46.8421052631579 4	56.3157894736842	52.6315789473396	45.2631578947368
AOT	47.3684210526316	45.2631578947368	46.8421052631579 4	38.9473684210526	41.5789473684211	42.1052631578947 04	48.4210526315789 5	44.2105263157895	46.8421052631579 4
BA	50.0	38.8888888888889	38.8888888888889	44.4444444444444	51.8518518518505	35.1851851851805	51.8518518518505	51.8518518518505	37.037037037037
BANPU	54.7368421052639	55.7894736842105	57.3684210526316 04	55.7894736842105	54.7368421052639	50.5263157894737	53.6842105263158	55.2631578947368	56.8421052631579 4
BBL	46.3157894736842	44.2105263157895	46.3157894736842	51.5789473684211 05	43.6842105263157 95	43.1578947368421	47.8947368421053	45.2631578947368	46.3157894736842
BCP	45.2631578947368	44.2105263157895	43.6842105263157 95	47.8947368421053	48.4210526315789 5	43.6842105263157 95	42.6315789473396	46.3157894736842	43.6842105263157 95
BDMS	52.6315789473396	48.9473684210526	44.2105263157895	46.3157894736842	47.3684210526316	42.1052631578947 04	54.7368421052639	47.3684210526316	44.2105263157895
BEC	47.8947368421053	48.4210526315789 5	53.6842105263158	47.3684210526316	50.5263157894737	47.8947368421053	47.3684210526316	50.5263157894737	53.1578947368421
BH	45.7894736842105	41.0526315789474	45.2631578947368	46.8421052631579 4	50.0	50.5263157894737	46.8421052631579 4	44.2105263157895	45.2631578947368
BLA	43.6842105263157 95	47.8947368421053	40.0	46.3157894736842	47.3684210526316	56.8421052631579 4	43.6842105263157 95	47.8947368421053	40.0
BTS	48.9473684210526	54.7368421052639	46.8421052631579 4	43.6842105263157 95	43.1578947368421	40.5263157894737	46.3157894736842	53.6842105263158	47.8947368421053
CBG	52.0	57.9999999999999	54.0	70.0	52.0	72.0	57.9999999999999	52.0	54.0
CENTEL	45.2631578947368	54.7368421052639	41.5789473684211	46.3157894736842	47.8947368421053	42.1052631578947 04	44.2105263157895	56.3157894736842	41.5789473684211
CK	46.8421052631579 4	51.5789473684211 05	45.2631578947368	47.3684210526316	50.5263157894737	42.6315789473396	48.4210526315789 5	51.5789473684211 05	45.2631578947368
CPALL	53.6842105263158	55.7894736842105	48.9473684210526	51.0526315789474 05	50.5263157894737	44.2105263157895	54.2105263157895	56.3157894736842	48.9473684210526
CPF	56.8421052631579 4	54.2105263157895	55.2631578947368	51.0526315789474 05	53.6842105263158	54.2105263157895	55.2631578947368	53.6842105263158	55.2631578947368
CPN	50.0	43.6842105263157 95	52.6315789473396	43.1578947368421	41.0526315789474	38.9473684210526	49.4736842105263	44.7368421052632	52.1052631578947
DELTA	54.2105263157895	52.6315789473396	41.5789473684211	51.0526315789474 05	51.5789473684211 05	42.6315789473396	52.1052631578947	52.1052631578947	41.5789473684211
DTAC	45.7894736842105	47.8947368421053	48.9473684210526	50.5263157894737	49.4736842105263	56.3157894736842	45.7894736842105	46.8421052631579 4	48.9473684210526
EGCO	45.7894736842105	50.5263157894737	45.7894736842105	48.9473684210526	47.3684210526316	46.8421052631579 4	46.8421052631579 4	51.5789473684211 05	45.7894736842105
GLOW	45.2631578947368	44.7368421052632	44.7368421052632	47.8947368421053	42.6315789473396	46.8421052631579 4	44.2105263157895	44.7368421052632	44.7368421052632
GPSC	43.75	50.0	62.5	43.75	43.75	43.75	56.25	56.25	62.5
HMPRO	48.4210526315789 5	47.3684210526316	46.8421052631579 4	53.1578947368421	54.2105263157895	51.5789473684211 05	48.9473684210526	47.3684210526316	46.8421052631579 4
INTUCH	48.4210526315789 5	48.4210526315789 5	45.2631578947368	55.7894736842105	53.6842105263158	45.7894736842105	47.8947368421053	48.4210526315789 5	44.7368421052632
IRPC	50.5263157894737	44.7368421052632	53.1578947368421	45.2631578947368	48.9473684210526	52.1052631578947	53.6842105263158	45.7894736842105	53.1578947368421
IVL	50.0	53.1578947368421	49.4736842105263	45.2631578947368	46.8421052631579 4	47.8947368421053	50.0	52.6315789473396	49.4736842105263
KBANK	45.2631578947368	46.8421052631579 4	47.8947368421053	50.0	47.3684210526316	47.3684210526316	45.2631578947368	46.8421052631579 4	47.8947368421053
KCE	52.6315789473396	51.5789473684211 05	54.2105263157895	46.3157894736842	45.2631578947368	39.4736842105263	52.6315789473396	51.5789473684211 05	53.6842105263158
KTB	51.0526315789474 05	53.6842105263158	51.5789473684211	52.1052631578947	52.6315789473396	40.5263157894737	50.0	50.5263157894737	51.5789473684211
LH	55.2631578947368	52.6315789473396	56.8421052631579 4	43.1578947368421	48.4210526315789 5	40.0	54.7368421052639	53.1578947368421	56.8421052631579 4
MINT	48.9473684210526	48.4210526315789 5	50.5263157894737	53.6842105263158	52.1052631578947	38.4210526315789	49.4736842105263	48.4210526315789 5	50.5263157894737
MTLS	40.8163265306122	46.9387755102041	42.8571428571429	53.0612244897959	48.9795918367347	44.8979591836735	40.8163265306122	44.8979591836735	42.8571428571429
PS	50.0	45.2631578947368	54.7368421052639	50.5263157894737	45.2631578947368	60.0	49.4736842105263	47.8947368421053	55.2631578947368
PTT	48.4210526315789 5	49.4736842105263	54.7368421052639	45.2631578947368	42.1052631578947 04	44.7368421052632	47.8947368421053	49.4736842105263	54.7368421052639
PTTEP	52.1052631578947	55.7894736842105	50.5263157894737	49.4736842105263	54.7368421052639	50.0	53.6842105263158	55.2631578947368	50.0
PTTGC	47.8947368421053	48.9473684210526	50.0	50.0	54.7368421052639	52.6315789473396	51.0526315789405	50.0	50.0
ROBINS	53.1578947368421	54.2105263157895	55.2631578947368	50.0	49.4736842105263	51.5789473684211 05	53.1578947368421	53.1578947368421	55.2631578947368
SAWAD	47.7777777777778	45.5555555555556	63.3333333333333	35.5555555555556	41.1111111111111	42.2222222222222	51.1111111111109	47.7777777777778	63.3333333333333
SCB	42.6315789473396	46.8421052631579 4	47.8947368421053	50.0	48.4210526315789 5	50.5263157894737	41.5789473684211	46.8421052631579 4	47.8947368421053
SCC	50.5263157894737	45.7894736842105	50.0	43.6842105263157 95	42.1052631578947 04	42.1052631578947 04	50.5263157894737	46.8421052631579 4	50.0
TASCO	49.4736842105263	47.3684210526316	42.6315789473396	48.4210526315789 5	46.3157894736842	46.3157894736842	49.4736842105263	48.4210526315789 5	42.1052631578947 04
TCAP	42.6315789473396	42.6315789473396	45.7894736842105	52.1052631578947	49.4736842105263	42.6315789473396	40.5263157894737	42.6315789473396	45.2631578947368
TMB	52.1052631578947	56.3157894736842	45.2631578947368	44.7368421052632	48.9473684210526	43.6842105263157 95	52.1052631578947	56.8421052631579 4	45.2631578947368
TOP	47.3684210526316	46.3157894736842	45.2631578947368	51.0526315789474 05	51.0526315789405	43.6842105263157 95	47.8947368421053	46.3157894736842	44.7368421052632
TPIPL	40.5263157894737	55.7894736842105	42.1052631578947 04	62.6315789473684	64.7368421052632 1	60.5263157894736 9	43.1578947368421	55.2631578947368	42.1052631578947 04
TRUE	46.3157894736842	49.4736842105263	53.1578947368421	53.6842105263158	51.5789473684211 05	52.6315789473396	46.8421052631579 4	49.4736842105263	53.1578947368421
TTW	44.7368421052632	45.2631578947368	39.4736842105263	40.0	36.8421052631579	39.4736842105263	44.7368421052632	45.2631578947368	39.4736842105263
TU	53.6842105263158	53.6842105263158	46.3157894736842	46.3157894736842	50.0	44.2105263157895	50.5263157894737	53.6842105263158	46.3157894736842
WHA	49.4736842105263	47.3684210526316	50.5263157894737	53.1578947368421	54.7368421052639	44.7368421052632	53.6842105263158	47.3684210526316	50.5263157894737

Table 86 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	56.3157894736842	55.7894736842105	45.7894736842105	51.052631578947405	55.26315789473684	47.89473684210526	52.631578947368396	55.26315789473684	45.7894736842105
AOT	48.9473684210526	50.52631578947368	46.31578947368421	41.052631578947405	41.052631578947405	38.94736842105263	41.052631578947405	50.0	46.31578947368421
BA	48.148148148148	42.592592592592595	38.888888888888889	50.0	53.703703703703695	53.703703703703695	50.0	42.592592592592595	38.888888888888889
BANPU	52.1052631578947	53.684210526315	57.368421052631604	55.7894736842105	55.26315789473684	52.1052631578947	53.157894736842	54.736842105263	57.368421052631604
BBL	48.9473684210526	45.7894736842105	44.7368421052632	50.0	48.9473684210526	45.7894736842105	50.52631578947368	45.7894736842105	43.684210526315795
BCP	44.7368421052632	44.210526315789	43.684210526315795	51.052631578947405	47.894736842105263	43.684210526315795	47.894736842105263	44.210526315789	43.684210526315795
BDMS	51.052631578947405	48.9473684210526	44.7368421052632	44.7368421052632	45.263157894736842	42.105263157894704	51.578947368421105	48.421052631578895	44.2105263157895
BEC	45.7894736842105	47.894736842105	52.1052631578947	47.36842105263316	50.52631578947368	50.0	46.315789473684	47.894736842105	51.578947368421105
BH	47.894736842105263	46.842105263157904	48.9473684210526	48.421052631578895	47.894736842105263	49.473684210526316	47.894736842105263	47.368421052631	48.9473684210526
BLA	46.315789473684	47.894736842105263	40.0	46.842105263157904	47.368421052631	42.105263157894704	42.105263157894704	47.894736842105263	40.0
BTS	51.052631578947405	52.631578947368396	46.315789473684	42.631578947368396	50.0	41.578947368421	51.052631578947405	50.0	46.315789473684
CBG	62.0	57.99999999999999	54.0	70.0	68.0	72.0	70.0	50.0	54.0
CENTEL	45.2631578947368	54.736842105263 19	42.631578947368396	45.2631578947368	50.0	42.105263157894704	44.7368421052632	56.315789473684 2	42.631578947368396
CK	45.2631578947368	50.0	41.052631578974	53.157894736842 1	49.473684210526 3	51.052631578947405	45.7894736842105	50.0	41.578947368421 1
CPALL	55.2631578947368	55.7894736842105	48.9473684210526	50.52631578947 37	50.52631578947 37	44.210526315789 5	54.210526315789 5	54.736842105263 19	48.9473684210526
CPF	50.52631578947 37	50.0	56.315789473684 2	51.052631578947405	55.26315789473684	57.894736842105296	52.1052631578947	48.421052631578895	55.7894736842105
CPN	51.052631578947405	45.263157894736 83	52.631578947368396	43.15789473684 21	41.052631578974	38.9473684210526	51.052631578947405	43.684210526315795	52.1052631578947
DELTA	52.631578947368396	51.052631578947405	41.578947368421 1	49.473684210526 3	51.578947368421105	42.631578947368396	52.631578947368396	51.578947368421105	41.578947368421 1
DTAC	48.421052631578895	48.9473684210526	50.0	46.842105263157904	46.842105263157904	52.631578947368396	47.368421052636	47.368421052636	50.0
EGCO	45.2631578947368	51.052631578947405	45.7894736842105	43.684210526315795	47.368421052636	46.842105263157904	46.315789473684 2	50.0	45.7894736842105
GLOW	48.9473684210526	45.7894736842105	44.736842105263 2	46.842105263157904	43.684210526315795	46.842105263157904	47.894736842105 3	46.842105263157904	44.736842105263 2
GPSC	43.75	50.0	62.5	43.75	43.75	43.75	56.25	50.0	62.5
HMPRO	48.421052631578895	48.421052631578895	48.421052631578895	52.631578947368396	54.210526315789 5	51.578947368421105	48.421052631578895	48.421052631578895	48.421052631578895
INTUCH	49.4736842105263	47.894736842105 3	45.263157894736 8	55.7894736842105	53.684210526315 8	45.7894736842105	47.368421052636	48.9473684210526	44.736842105263 2
IRPC	52.1052631578947	47.368421052636	53.684210526315 8	48.421052631578895	50.0	49.473684210526 3	51.052631578947405	48.947368421052 6	53.684210526315 8
IVL	52.1052631578947	51.052631578947405	51.052631578947405	50.52631578947 37	50.0	46.842105263157904	48.9473684210526	51.052631578947405	51.052631578947405
KBANK	48.421052631578895	44.210526315789 5	48.947368421026	45.7894736842105	47.894736842105 3	48.947368421026	47.894736842105 3	43.684210526315795	48.947368421026
KCE	52.1052631578947	50.52631578947 37	53.157894736842 1	44.736842105263 2	44.736842105263 2	40.0	48.421052631578895	51.578947368421105	53.157894736842 1
KTB	50.0	52.1052631578947	41.578947368421 1	53.157894736842 1	52.631578947368396	40.52631578947 37	47.368421052636	52.631578947368396	41.578947368421 1
LH	58.9473684210526	54.736842105263 19	56.842105263157904	44.210526315789 5	45.789473684210 5	46.315789473684 2	61.578947368421 1	54.2105263157895	57.368421052631604
MINT	50.0	48.421052631578895	41.052631578974	48.947368421026	45.789473684210 5	42.105263157894704	48.421052631578895	49.473684210526 3	39.473684210526 3
MTLS	40.8163265306122	44.8979591836735	42.8571428571429	48.9795918367347	44.8979591836735	44.8979591836735	42.8571428571429	46.9387755102041	42.8571428571429
PS	47.368421052636 16	47.894736842105 3	58.947368421026	56.842105263157904	53.157894736842 1	60.526315789473694	48.421052631578895	52.631578947368396	58.947368421026
PTT	49.4736842105263	49.473684210526 3	54.736842105263 19	45.263157894736 8	43.684210526315795	44.736842105263 2	46.315789473684 2	48.947368421026	54.736842105263 19
PTTEP	54.736842105263 19	55.789473684210 5	54.736842105263 19	47.368421052636	45.789473684210 5	52.1052631578947	53.684210526315 8	55.789473684210 5	54.736842105263 19
PTTGC	52.1052631578947	49.473684210526 3	50.0	50.52631578947 37	54.736842105263 19	52.631578947368396	46.842105263157904	50.52631578947 37	50.0
ROBINS	52.1052631578947	54.210526315789 5	55.263157894736 8	52.1052631578947	50.0	51.578947368421105	53.157894736842 1	53.157894736842 1	55.263157894736 8
SAWAD	51.111111111111109	46.666666666667	63.333333333333	34.444444444444 4	38.888888888889	42.222222222222 2	52.222222222222 2	46.666666666667	63.333333333333 3
SCB	41.5789473684211	45.789473684210 5	47.894736842105 3	47.894736842105 3	47.894736842105 3	50.52631578947 37	42.105263157894704	46.842105263157904	47.894736842105 3
SCC	50.0	46.842105263157904	50.0	44.736842105263 2	40.52631578947 37	43.157894736842 1	50.0	47.368421052636	50.52631578947 37
TASCO	47.894736842105 3	49.473684210526 3	45.789473684210 5	48.947368421052 6	45.263157894736 8	45.263157894736 8	47.894736842105 3	50.0	45.263157894736 8
TCAP	42.631578947368396	42.631578947368396	45.789473684210 5	52.631578947368396	50.52631578947 37	42.631578947368396	42.631578947368396	42.631578947368396	45.263157894736 8
TMB	53.684210526315 8	56.315789473684 2	45.263157894736 8	45.789473684210 5	49.473684210526 3	43.684210526315795	53.684210526315 8	56.315789473684 2	45.263157894736 8
TOP	48.421052631578895	45.789473684210 5	45.263157894736 8	51.052631578947405	52.10526315789 47	43.684210526315795	46.315789473684 2	46.842105263157904	44.736842105263 2
TPIPL	49.4736842105263	53.684210526315 8	41.052631578974	61.578947368421 1	64.210526315789 5	47.368421052636 16	49.473684210526 3	53.684210526315 8	41.052631578974
TRUE	49.473684210526 3	50.526315789473 7	53.157894736842 1	46.315789473684 2	45.263157894736 8	55.263157894736 8	51.052631578947405	50.526315789473 7	53.157894736842 1
TTW	45.789473684210 5	46.842105263157904	39.473684210526 3	41.052631578974	36.315789473684 2	38.947368421052 6	46.842105263157904	46.842105263157904	39.473684210526 3
TU	52.1052631578947	53.684210526315 8	46.315789473684 2	49.473684210526 3	52.10526315789 47	44.210526315789 5	55.263157894736 8	53.157894736842 1	46.315789473684 2
WHA	50.0	45.789473684210 5	53.157894736842 1	54.736842105263 19	56.315789473684 2	47.894736842105 3	52.1052631578947	47.368421052636 16	53.157894736842 1

Table 87 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	46.3157894736842	47.894736842105 3	43.684210526315795	45.7894736842105	47.368421052636 16	52.631578947368396	44.210526315789 5	47.894736842105 3	43.684210526315795
AOT	39.4736842105263	41.052631578974	38.947368421026	52.1052631578947	48.947368421026	45.789473684210 5	44.736842105263 2	41.052631578974	38.947368421026
BA	50.0	35.185185185185205	38.888888888889	38.888888888889	35.185185185185205	38.888888888889	38.888888888889	35.185185185185205	38.888888888889
BANPU	42.631578947368396	48.421052631578895	41.578947368421 1	47.894736842105 3	53.684210526315 8	42.631578947368396	45.789473684210 5	51.578947368421105	41.578947368421 1
BBL	45.789473684210 5	46.842105263157904	44.736842105263 2	55.789473684210 5	52.631578947368396	54.210526315789 5	46.842105263157904	46.842105263157904	44.736842105263 2
BCP	37.368421052631604	44.736842105263 2	38.421052631578 9	47.894736842105 3	47.368421052636 16	55.789473684210 5	37.368421052631604	41.052631578974	38.421052631578 9
BDMS	43.157894736842 1	43.684210526315795	37.368421052631604	45.789473684210 5	45.789473684210 5	47.368421052636 16	43.157894736842 1	40.526315789473 7	37.368421052631604
BEC	41.578947368421 1	46.315789473684 2	43.684210526315795	44.736842105263 2	50.526315789473 7	49.473684210526 3	42.631578947368396	47.894736842105 3	43.684210526315795
BH	50.0	47.368421052636 16	49.473684210526 3	42.631578947368396	45.789473684210 5	47.368421052636 16	47.368421052636 16	45.789473684210 5	49.473684210526 3
BLA	47.894736842105 3	49.473684210526 3	45.789473684210 5	47.894736842105 3	52.10526315789 47	63.157894736842 1	42.631578947368396	49.473684210526 3	45.789473684210 5
BTS	44.736842105263 2	41.052631578974	42.105263157894704	54.210526315789 5	43.684210526315795	48.947368421052 6	42.631578947368396	43.157894736842 1	42.631578947368396
CBG	48.0	48.0	32.0	54.0	56.000000000000 01	48.0	48.0	48.0	32.0
CENTEL	41.052631578974	43.157894736842 1	38.947368421052 6	51.052631578947405	45.263157894736 8	42.105263157894704	41.578947368421 1	40.526315789473 7	38.947368421052 6
CK	39.473684210526 3	39.473684210526 3	38.947368421052 6	44.736842105263 2	47.894736842105 3	42.105263157894704	38.947368421052 6	39.473684210526 3	38.947368421052 6
CPALL	41.578947368421 1	41.578947368421 1	43.157894736842 1	43.157894736842 1	44.736842105263 2	53.684210526315 8	42.105263157894704	40.526315789473 7	43.157894736842 1
CPF	47.894736842105 3	47.368421052636 16	43.684210526315795	51.052631578947405	50.526315789473 7	55.789473684210 5	48.421052631578895	47.368421052636 16	43.684210526315795
CPN	40.526315789473 7	36.315789473684 2	44.736842105263 2	54.736842105263 19	52.631578947368396	45.263157894736 8	43.157894736842 1	37.368421052631604	44.736842105263 2
DELTA	48.947368421026	48.947368421026	45.263157894736 8	53.157894736842 1	55.263157894736 8	48.947368421026	48.421052631578895	47.894736842105 3	45.263157894736 8
DTAC	45.789473684210 5	45.263157894736 8	50.0	56.315789473684 2	44.736842105263 2	52.631578947368396	46.315789473684 2	47.894736842105 3	50.0
EGCO	39.473684210526 3	41.052631578974	38.947368421026	48.421052631578895	45.789473684210 5	53.157894736842 1	38.421052631578 9	41.052631578974	38.947368421026
GLOW	44.210526315789 5	46.315789473684 2	40.526315789473 7	47.894736842105 3	50.526315789473 7	51.578947368421105	45.263157894736 8	44.736842105263 2	40.526315789473 7
GPSC	37.5	43.75	43.75	56.25	56.25	43.75	37.5	43.75	43.75
HMPRO	37.368421052631604	39.473684210526 3	36.842105263159	46.315789473684 2	45.263157894736 8	46.315789473684 2	36.842105263159	40.0	36.842105263159
INTUCH	42.105263157894704	47.368421052636 16	42.105263157894704	40.526315789473 7	47.368421052636 16	46.842105263157904	43.157894736842 1	45.263157894736 8	42.105263157894704
IRPC	46.842105263157904	48.421052631578895	43.157894736842 1	49.473684210526 3	47.894736842103	52.1052631578947	46.315789473684 2	49.473684210526 3	43.157894736842 1
IVL	41.052631578974	40.0	37.894736842105296	42.105263157894704	52.105263157894 7	41.052631578974	41.052631578974	40.0	37.894736842105296
KBANK	42.631578947368396	46.315789473684 2	45.263157894736 8	52.1052631578947	50.526315789473 7	50.526315789473 7	43.157894736842 1	46.315789473684 2	45.263157894736 8
KCE	43.157894736842 1	45.263157894736 8	40.0	49.473684210526 3	46.842105263157904	56.842105263157904	43.684210526315795	43.684210526315795	40.0
KTB	40.526315789473 7	44.736842105263 2	39.473684210526 3	52.631578947368396	45.263157894736 8	46.842105263157904	41.578947368421 1	42.105263157894704	39.473684210526 3
LH	41.578947368421 1	45.263157894736 8	38.421052631578 9	46.842105263157904	50.0	48.421052631578895	41.578947368421 1	44.210526315789 5	38.421052631578 9
MINT	38.947368421026	48.421052631578895	42.105263157894704	51.578947368421105	46.315789473684 2	51.578947368421105	38.947368421026	47.368421052636 16	42.105263157894704
MTLS	48.9795918367347	48.9795918367347	48.9795918367347	51.020408163265294	51.020408163265294	48.9795918367347	51.020408163265294	48.9795918367347	48.9795918367347
PS	36.315789473684 2	38.947368421026	36.315789473684 2	48.421052631578895	45.263157894736 8	37.894736842105296	36.315789473684 2	40.526315789473 7	36.315789473684 2
PTT	40.526315789473 7	37.894736842105296	41.052631578974	50.0	44.210526315789 5	57.368421052631604	39.473684210526 3	40.526315789473 7	41.052631578974
PTTEP	56.315789473684 2	51.052631578947405	55.263157894736 8	50.0	48.947368421026	53.157894736842 1	54.210526315789 5	50.526315789473 7	55.263157894736 8
PTTGC	46.315789473684 2	47.894736842105 3	46.842105263157904	44.736842105263 2	46.315789473684 2	53.157894736842 1	45.789473684210 5	46.842105263157904	46.842105263157904
ROBINS	45.789473684210 5	46.842105263157904	37.894736842105296	54.210526315789 5	50.526315789473 7	47.368421052636 16	46.842105263157904	46.842105263157904	37.894736842105296
SAWAD	53.333333333333	52.222222222222 2	41.111111111111 1	55.555555555556	54.444444444439	45.555555555556	50.0	53.333333333333	41.111111111111 1
SCB	42.105263157894704	43.684210526315795	41.052631578974	44.736842105263 2	47.894736842105 3	50.526315789473 7	42.631578947368396	40.0	41.052631578974
SCC	41.052631578974	40.0	40.0	54.210526315789 5	47.894736842105 3	54.210526315789 5	41.578947368421 1	41.578947368421 1	40.0
TASCO	45.789473684210 5	47.894736842105 3	46.315789473684 2	48.947368421052 6	52.631578947368396	56.315789473684 2	50.0	48.421052631578895	46.315789473684 2
TCAP	38.421052631578 9	37.894736842105296	38.421052631578 9	50.526315789473 7	50.526315789473 7	50.0	40.0	37.368421052631604	38.421052631578 9
TMB	43.684210526315795	42.105263157894704	43.684210526315795	53.684210526315 8	43.684210526315795	48.947368421026	43.684210526315795	42.105263157894704	43.684210526315795
TOP	39.473684210526 3	40.526315789473 7	43.157894736842 1	46.842105263157904	49.473684210526 3	44.210526315789 5	40.526315789473 7	40.526315789473 7	43.157894736842 1
TPIPL	43.684210526315795	44.210526315789 5	44.210526315789 5	46.315789473684 2	49.473684210526 3	48.421052631578895	44.736842105263 2	44.210526315789 5	44.210526315789 5
TRUE	41.578947368421 1	41.578947368421 1	45.263157894736 8	54.736842105263 19	52.105263157894 7	48.421052631578895	40.526315789473 7	43.157894736842 1	45.263157894736 8
TTW	32.631578947368 4	36.842105263159	34.736842105263 2	52.105263157894 7	45.263157894736 8	40.526315789473 7	34.736842105263 2	36.842105263159	34.736842105263 2
TU	39.473684210526 3	42.105263157894704	40.0	53.157894736842 1	48.421052631578895	55.263157894736 8	39.473684210526 3	44.210526315789 5	40.0
WHA	38.421052631578 9	39.473684210526 3	39.473684210526 3	48.947368421026	43.157894736842 1	42.631578947368396	36.842105263159	39.473684210526 3	39.473684210526 3

PredictDirection10 : ADX (Average Directional Index), DIP (Directional Index Plus), and DIM (Directional Index Minus)

Table 88 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	44.73684210526316	43.684210526315795	43.684210526315795	44.21052631578947	43.684210526315795	43.684210526315795	44.73684210526316
AOT	59.4736842105263	60.0	58.421052631578995	59.4736842105263	60.0	58.421052631578995	59.4736842105263	60.0	57.894736842105296
BA	44.44444444444444	50.0	50.0	50.0	51.851851851851805	59.25925925925931	48.148148148148	50.0	50.0
BANPU	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.05263157894474	41.57894736842111	41.05263157894474
BBL	44.73684210526316	44.21052631578947	42.105263157894704	43.684210526315795	45.78947368421053	41.05263157894474	43.157894736842105	44.73684210526316	41.57894736842111
BCP	49.4736842105263	52.10526315789474	53.157894736842105	50.0	48.94736842105263	51.578947368421105	51.578947368421105	53.157894736842105	53.157894736842105
BDMS	60.526315789473694	61.052631578947405	60.0	60.526315789473694	60.526315789473694	60.526315789473694	60.526315789473694	61.052631578947405	60.0
BEC	45.26315789473684	45.26315789473684	45.78947368421053	45.26315789473684	45.26315789473684	45.78947368421053	45.26315789473684	45.26315789473684	45.26315789473684
BH	52.10526315789474	54.21052631578947	55.263157894736805	52.631578947368396	53.68421052631579	54.21052631578947	52.631578947368396	53.68421052631579	54.736842105263194
BLA	48.421052631578895	47.89473684210526	47.36842105263158	47.89473684210526	47.36842105263158	46.31578947368421	48.421052631578895	48.421052631578895	47.89473684210526
BTS	52.10526315789474	52.10526315789474	52.10526315789474	50.0	50.0	52.10526315789474	53.157894736842105	52.10526315789474	52.10526315789474
CBG	70.0	70.0	68.0	70.0	70.0	70.0	70.0	70.0	68.0
CENTEL	56.31578947368421	56.31578947368421	57.368421052631604	57.368421052631604	56.842105263157904	57.894736842105296	56.31578947368421	56.31578947368421	57.368421052631604
CK	52.631578947368396	53.68421052631579	53.157894736842105	51.578947368421105	53.157894736842105	53.157894736842105	52.10526315789474	52.631578947368396	53.157894736842105
CPALL	56.31578947368421	55.789473684210526	52.631578947368396	55.789473684210526	57.368421052631604	53.68421052631579	55.789473684210526	56.842105263157904	52.631578947368396
CPF	48.421052631578895	50.0	49.473684210526316	50.0	49.473684210526316	48.421052631578895	48.947368421052626	49.473684210526316	49.473684210526316
CPN	54.736842105263194	54.736842105263194	55.263157894736805	52.631578947368396	53.68421052631579	52.10526315789474	55.263157894736805	54.736842105263194	55.263157894736805
DELTA	50.526315789473737	53.157894736842105	52.631578947368396	53.157894736842105	53.157894736842105	53.68421052631579	48.421052631578895	53.157894736842105	52.631578947368396
DTAC	50.0	50.0	51.052631578947405	50.0	50.0	50.526315789473737	50.0	50.0	51.052631578947405
EGCO	54.736842105263194	56.842105263157904	56.31578947368421	56.842105263157904	56.842105263157904	56.842105263157904	55.263157894736805	57.894736842105296	55.789473684210526
GLOW	51.578947368421105	47.89473684210526	50.0	50.0	46.842105263157904	47.89473684210526	51.578947368421105	47.89473684210526	49.473684210526316
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	60.0	60.526315789473694	57.894736842105296	58.421052631578995	60.0	57.894736842105296	58.421052631578995	60.0	57.894736842105296
INTUCH	40.526315789473737	41.05263157894474	41.05263157894474	40.526315789473737	42.105263157894704	40.526315789473737	40.526315789473737	41.05263157894474	41.05263157894474
IRPC	55.789473684210526	55.789473684210526	56.842105263157904	55.263157894736805	55.789473684210526	55.789473684210526	55.789473684210526	55.263157894736805	56.842105263157904
IVL	54.736842105263194	56.842105263157904	54.21052631578947	52.631578947368396	58.421052631578995	53.68421052631579	55.263157894736805	56.31578947368421	54.21052631578947
KBANK	45.26315789473684	45.789473684210526	43.157894736842105	43.684210526315795	47.89473684210526	44.21052631578947	45.789473684210526	46.842105263157904	43.684210526315795
KCE	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
KTB	46.31578947368421	43.684210526315795	43.157894736842105	43.157894736842105	42.631578947368396	44.736842105263194	45.789473684210526	44.21052631578947	43.157894736842105
LH	43.684210526315795	43.157894736842105	49.473684210526316	45.26315789473684	43.157894736842105	48.421052631578895	40.526315789473737	41.05263157894474	49.473684210526316
MINT	58.421052631578995	58.421052631578995	57.894736842105296	57.894736842105296	58.421052631578995	58.421052631578995	58.421052631578995	57.894736842105296	57.894736842105296
MTLS	55.102040816326594	51.020408163265294	51.020408163265294	51.020408163265294	46.938775510204104	48.97959183673469	53.061224489795904	51.020408163265294	51.020408163265294
PS	52.631578947368396	52.631578947368396	58.947368421052626	55.263157894736805	52.631578947368396	55.789473684210526	53.157894736842105	52.10526315789474	58.947368421052626
PTT	44.21052631578947	44.21052631578947	42.631578947368396	42.105263157894704	43.157894736842105	43.157894736842105	45.26315789473684	44.21052631578947	42.631578947368396
PTTEP	48.947368421052626	47.89473684210526	47.89473684210526	48.421052631578895	49.473684210526316	48.947368421052626	50.0	47.89473684210526	47.89473684210526
PTTGC	54.21052631578947	54.736842105263194	54.21052631578947	53.684210526315795	55.263157894736805	54.736842105263194	54.21052631578947	55.263157894736805	54.736842105263194
ROBINS	52.631578947368396	52.10526315789474	53.684210526315795	54.21052631578947	51.052631578947405	53.684210526315795	53.157894736842105	51.578947368421105	53.684210526315795
SAWAD	47.77777777777778	50.0	50.0	48.88888888888889	51.111111111111094	56.666666666666714	43.333333333333314	48.88888888888889	50.0
SCB	42.105263157894704	39.473684210526315	43.157894736842105	40.0	38.947368421052626	42.105263157894704	42.105263157894704	39.473684210526315	43.157894736842105
SCC	53.684210526315795	52.631578947368396	55.789473684210526	53.684210526315795	52.10526315789474	53.684210526315795	53.157894736842105	52.631578947368396	55.789473684210526
TASCO	51.052631578947405	52.10526315789474	49.473684210526316	50.526315789473737	52.10526315789474	45.26315789473684	51.052631578947405	51.578947368421105	48.947368421052626
TCAP	52.631578947368396	53.157894736842105	53.157894736842105	51.578947368421105	53.157894736842105	54.736842105263194	52.631578947368396	53.157894736842105	53.157894736842105
TMB	52.631578947368396	54.21052631578947	50.526315789473737	53.684210526315795	53.684210526315795	52.631578947368396	53.157894736842105	54.21052631578947	50.526315789473737
TOP	60.526315789473694	60.526315789473694	60.526315789473694	60.526315789473694	60.526315789473694	60.0	60.526315789473694	60.526315789473694	60.526315789473694
TPIPL	54.736842105263194	57.368421052631604	56.31578947368421	49.473684210526316	54.736842105263194	55.263157894736805	54.736842105263194	57.368421052631604	56.31578947368421
TRUE	48.947368421052626	50.0	52.631578947368396	47.89473684210526	48.947368421052626	51.052631578947405	48.947368421052626	48.947368421052626	52.631578947368396
TTW	57.894736842105296	51.578947368421105	56.842105263157904	57.368421052631604	52.10526315789474	55.263157894736805	58.947368421052626	51.578947368421105	55.789473684210526
TU	51.578947368421105	50.526315789473737	52.631578947368396	54.736842105263194	50.0	51.578947368421105	52.10526315789474	51.578947368421105	52.631578947368396
WHA	54.736842105263194	53.684210526315795	61.052631578947405	54.736842105263194	54.21052631578947	55.789473684210526	55.789473684210526	53.684210526315795	60.526315789473694

Table 89 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	51.578947368421105	49.4736842105263	49.4736842105263	48.421052631578895	47.3684210526316	53.15789473684211	47.89473684210526	49.4736842105263	49.4736842105263
AOT	45.26315789473684	49.4736842105263	45.7894736842105	41.05263157894474	44.736842105263	43.684210526315795	44.736842105263	49.4736842105263	45.7894736842105
BA	48.148148148148	40.7407407407407	40.7407407407407	38.88888888888889	37.037037037037	37.037037037037	50.0	40.7407407407407	40.7407407407407
BANPU	53.68421052631578	50.0	53.68421052631578	52.1052631578947	52.1052631578947	45.26315789473684	53.68421052631578	48.9473684210526	53.68421052631578
BBL	44.7368421052632	48.421052631578895	43.1578947368421	47.8947368421053	48.9473684210526	43.15789473684211	44.21052631578947	47.3684210526316	44.21052631578947
BCP	49.473684210526	47.3684210526316	50.52631578947437	43.684210526315795	43.15789473684211	43.684210526315795	50.52631578947437	48.421052631578895	50.0
BDMS	51.052631578947405	52.1052631578947	49.4736842105263	46.3157894736842	47.3684210526316	47.3684210526316	47.89473684210526	53.68421052631578	48.421052631578895
BEC	56.31578947368421	52.631578947368396	54.21052631578947	55.26315789473684	52.631578947368396	48.9473684210526	57.368421052631604	53.68421052631578	54.21052631578947
BH	47.3684210526316	49.4736842105263	48.9473684210526	49.4736842105263	50.52631578947437	51.578947368421105	47.3684210526316	49.4736842105263	48.421052631578895
BLA	48.9473684210526	45.7894736842105	44.21052631578947	51.578947368421105	52.631578947368396	56.842105263157904	48.9473684210526	45.7894736842105	44.21052631578947
BTS	48.9473684210526	51.052631578947405	49.4736842105263	43.684210526315795	40.52631578947437	39.473684210526	47.3684210526316	51.052631578947405	49.4736842105263
CBG	48.0	57.99999999999999	52.0	68.0	68.0	66.0	50.0	57.99999999999999	52.0
CENTEL	46.31578947368421	40.52631578947437	42.105263157894704	47.3684210526316	45.263157894736	43.15789473684211	46.31578947368421	41.05263157894474	42.631578947368396
CK	43.15789473684211	47.89473684210553	43.684210526315795	53.15789473684211	52.631578947368396	49.473684210526	44.736842105263	47.89473684210553	44.21052631578947
CPALL	53.684210526315	55.26315789473684	48.9473684210526	48.9473684210526	53.684210526315	51.578947368421105	53.684210526315	55.26315789473684	48.9473684210526
CPF	55.789473684210	54.73684210526319	54.2105263157895	52.631578947368396	50.0	54.73684210526319	55.789473684210	54.73684210526319	54.73684210526319
CPN	56.842105263157904	55.789473684210	56.31578947368421	52.631578947368396	50.526315789473	38.9473684210526	53.15789473684211	55.789473684210	56.31578947368421
DELTA	51.578947368421105	48.421052631578895	47.8947368421053	48.421052631578895	45.789473684210	45.26315789473684	51.052631578947405	47.3684210526316	47.3684210526316
DTAC	53.684210526315	51.578947368421105	51.052631578947405	52.631578947368396	51.052631578947405	51.052631578947405	53.684210526315	51.052631578947405	51.052631578947405
EGCO	51.052631578947405	45.26315789473684	46.842105263157904	43.684210526315795	46.31578947368421	50.0	50.526315789473	46.31578947368421	46.842105263157904
GLOW	46.31578947368421	44.736842105263	42.105263157894704	49.473684210526	47.89473684210553	47.368421052631	46.842105263157904	44.736842105263	42.105263157894704
GPSC	50.0	62.5	56.25	43.75	43.75	43.75	62.5	62.5	62.5
HMPRO	51.578947368421105	45.789473684210	51.578947368421105	44.736842105263	41.578947368421	44.736842105263	51.578947368421105	45.789473684210	51.052631578947405
INTUCH	50.0	50.526315789473	43.1578947368421	45.263157894736	46.842105263157904	46.842105263157904	49.473684210526	50.0	43.1578947368421
IRPC	53.15789473684211	56.842105263157904	53.1578947368421	48.9473684210526	45.789473684210	47.368421052631	51.578947368421105	56.842105263157904	53.1578947368421
IVL	44.7368421052632	46.31578947368421	54.2105263157895	46.842105263157904	45.263157894736	50.0	44.21052631578947	46.31578947368421	54.2105263157895
KBANK	56.31578947368421	54.21052631578947	58.421052631578895	45.263157894736	42.105263157894704	48.421052631578895	56.842105263157904	54.21052631578947	58.9473684210526
KCE	43.684210526315795	43.15789473684211	48.421052631578895	44.736842105263	46.842105263157904	43.684210526315795	44.21052631578947	41.57894736842	47.8947368421053
KTB	47.36842105263	43.15789473684211	47.8947368421053	55.789473684210	60.0	56.31578947368421	47.36842105263	43.15789473684211	47.8947368421053
LH	51.052631578947405	55.26315789473684	53.68421052631578	48.421052631578895	48.421052631578895	44.736842105263	51.052631578947405	54.73684210526319	53.1578947368421
MINT	53.684210526315	50.526315789473	47.36842105263	52.631578947368396	53.15789473684211	48.421052631578895	53.684210526315	51.578947368421105	46.842105263157904
MTLS	48.9795918367347	53.06122448979591	48.9795918367347	48.9795918367347	44.89795918636735	48.9795918367347	51.020408163265294	53.06122448979591	51.020408163265294
PS	55.789473684210	51.578947368421105	45.789473684210	51.578947368421105	48.947368421052	45.789473684210	55.789473684210	54.21052631578947	46.842105263157904
PTT	44.7368421052632	38.9473684210526	53.68421052631578	43.1578947368421	45.789473684210	42.631578947368396	43.684210526315795	39.4736842105263	53.1578947368421
PTTEP	55.26315789473684	53.684210526315	53.1578947368421	51.052631578947405	52.1052631578947	53.15789473684211	55.26315789473684	53.684210526315	53.1578947368421
PTTGC	53.15789473684211	55.789473684210	54.2105263157895	55.26315789473684	56.842105263157904	55.789473684210	52.1052631578947	55.26315789473684	53.1578947368421
ROBINS	57.894736842105296	57.368421052631604	55.789473684210	56.31578947368421	55.26315789473684	52.1052631578947	57.368421052631604	57.368421052631604	55.26315789473684
SAWAD	58.88888888888889	58.88888888888889	44.44444444444444	55.55555555555556	54.44444444444439	58.88888888888889	58.88888888888889	58.88888888888889	44.44444444444444
SCB	44.736842105263	45.789473684210	53.15789473684211	49.473684210526	52.631578947368396	53.684210526315	43.684210526315795	45.789473684210	51.578947368421105
SCC	49.473684210526	47.89473684210553	44.7368421052632	47.368421052631	43.684210526315795	44.21052631578947	50.0	46.842105263157904	45.26315789473684
TASCO	44.7368421052632	45.789473684210	47.368421052631	45.789473684210	45.789473684210	46.31578947368421	44.7368421052632	45.263157894736	46.842105263157904
TCAP	46.31578947368421	45.263157894736	43.684210526315795	44.736842105263	44.210526315789	41.57894736842211	46.31578947368421	46.842105263157904	43.684210526315795
TMB	48.9473684210526	50.526315789473	46.842105263157904	45.263157894736	42.631578947368396	44.21052631578947	49.473684210526	50.526315789473	47.368421052631
TOP	47.368421052631	47.89473684210553	50.0	46.31578947368421	44.210526315789	42.105263157894704	50.526315789473	48.421052631578895	50.0
TPIPL	47.368421052631	47.89473684210553	42.631578947368396	45.789473684210	48.9473684210526	61.052631578947405	47.89473684210553	50.0	42.631578947368396
TRUE	53.684210526315	51.578947368421105	52.631578947368396	48.947368421052	52.1052631578947	48.421052631578895	51.578947368421105	51.578947368421105	52.631578947368396
TTW	40.526315789473	44.21052631578947	40.526315789473	39.473684210526	42.105263157894704	42.631578947368396	44.736842105263	43.1578947368421	41.05263157894474
TU	45.789473684210	45.789473684210	44.21052631578947	52.1052631578947	50.526315789473	48.421052631578895	45.263157894736	44.736842105263	44.736842105263
WHA	50.0	48.947368421052	49.4736842105263	50.526315789473	51.578947368421105	45.789473684210	50.526315789473	48.421052631578895	49.4736842105263

Table 90 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	54.21052631578947	50.526315789473	51.578947368421105	50.0	47.89473684210553	53.15789473684211	54.736842105263	50.0	52.1052631578947
AOT	51.578947368421105	53.684210526315	49.473684210526	46.315789473684	45.263157894736	42.631578947368396	48.947368421052	52.631578947368396	49.473684210526
BA	51.851851851851	44.44444444444444	35.185185185185205	53.703703703703695	48.148148148148	46.296296296296305	48.148148148148	42.592592592595	37.037037037037
BANPU	52.1052631578947	48.947368421052	51.578947368421105	51.578947368421105	51.578947368421105	46.3157894736842	51.578947368421105	48.947368421052	52.1052631578947
BBL	42.631578947368396	49.4736842105263	44.21052631578947	47.8947368421053	46.315789473684	44.21052631578947	43.684210526315795	46.315789473684	43.684210526315795
BCP	48.421052631578895	47.36842105263	50.526315789473	43.1578947368421	42.631578947368396	43.684210526315795	50.0	47.89473684210553	50.0
BDMS	49.4736842105263	52.1052631578947	49.4736842105263	45.263157894736	47.89473684210553	47.3684210526316	48.947368421052	52.631578947368396	48.421052631578895
BEC	53.684210526315	53.684210526315	54.21052631578947	54.736842105263	53.15789473684211	52.1052631578947	52.631578947368396	54.21052631578947	54.21052631578947
BH	49.4736842105263	47.36842105263	47.36842105263	47.36842105263	46.842105263157904	49.4736842105263	48.947368421052	46.842105263157904	47.36842105263
BLA	47.89473684210553	45.263157894736	44.21052631578947	50.526315789473	51.578947368421105	56.842105263157904	48.421052631578895	44.736842105263	44.21052631578947
BTS	47.36842105263	51.052631578947405	49.4736842105263	41.05263157894474	42.105263157894704	42.631578947368396	49.4736842105263	51.578947368421105	49.4736842105263
CBG	52.0	52.0	52.0	66.0	68.0	66.0	50.0	57.99999999999999	52.0
CENTEL	46.31578947368421	41.57894736842211	43.684210526315795	47.36842105263	44.210526315789	43.15789473684211	45.263157894736	41.57894736842211	43.684210526315795
CK	44.736842105263	46.315789473684	42.105263157894704	47.89473684210553	49.473684210526	51.578947368421105	44.736842105263	45.789473684210	42.631578947368396
CPALL	53.684210526315	56.315789473684	48.947368421052	47.894736842105	53.15789473684211	51.578947368421105	53.684210526315	56.315789473684	48.947368421052
CPF	53.684210526315	55.789473684210	55.263157894736	49.473684210526	52.631578947368396	51.578947368421105	55.263157894736	55.789473684210	55.263157894736
CPN	57.368421052631604	55.789473684210	56.315789473684	53.684210526315	49.473684210526	38.9473684210526	57.368421052631604	55.789473684210	56.315789473684
DELTA	51.052631578947405	47.36842105263	47.89473684210553	47.36842105263	46.842105263157904	45.263157894736	50.0	47.36842105263	47.36842105263
DTAC	52.1052631578947	52.1052631578947	49.473684210526	51.052631578947405	47.36842105263	51.578947368421105	54.210526315789	51.052631578947405	49.473684210526
EGCO	50.0	44.736842105263	46.842105263157904	42.631578947368396	46.842105263157904	50.0	50.526315789473	44.736842105263	46.842105263157904
GLOW	47.36842105263	44.736842105263	43.1578947368421	48.947368421052	47.89473684210553	47.368421052631	46.842105263157904	44.736842105263	43.1578947368421
GPSC	56.25	62.5	56.25	43.75	43.75	43.75	56.25	62.5	62.5
HMPRO	43.157894736842	46.315789473684	51.052631578947405	48.947368421052	42.631578947368396	44.736842105263	44.210526315789	45.789473684210	50.526315789473
INTUCH	50.0	50.526315789473	43.1578947368421	44.210526315789	47.36842105263	46.842105263157904	49.473684210526	51.578947368421105	43.1578947368421
IRPC	52.1052631578947	56.315789473684	53.684210526315	48.947368421052	46.842105263157904	50.0	50.0	56.842105263157904	53.684210526315
IVL	51.052631578947405	46.315789473684	52.1052631578947	53.157894736842	47.89473684210553	53.15789473684211	51.052631578947405	46.315789473684	52.1052631578947
KBANK	52.631578947368396	51.052631578947405	55.263157894736	51.052631578947405	47.36842105263	42.631578947368396	52.1052631578947	51.052631578947405	55.263157894736
KCE	41.57894736842211	43.15789473684211	48.421052631578895	45.263157894736	43.684210526315795	42.631578947368396	42.105263157894704	43.15789473684211	48.421052631578895
KTB	45.263157894736	42.631578947368396	47.89473684210553	56.842105263157904	57.368421052631604	56.315789473684	47.89473684210553	43.15789473684211	48.421052631578895
LH	54.736842105263	50.0	54.2105263157895	51.052631578947405	53.157894736842	48.421052631578895	54.2105263157895	50.0	53.684210526315
MINT	46.31578947368421	45.789473684210	43.15789473684211	52.631578947368396	49.473684210526	44.736842105263	48.947368421052	46.315789473684	43.15789473684211
MTLS	48.9795918367347	53.06122448979591	48.9795918367347	48.9795918367347	44.89795918636735	48.9795918367347	51.020408163265294	48.9795918367347	51.020408163265294
PS	56.842105263157904	51.578947368421105	50.0	52.1052631578947	53.157894736842	45.263157894736	57.894736842105296	52.1052631578947	50.526315789473
PTT	42.105263157894704	39.473684210526	53.684210526315	43.1578947368421	46.315789473684	42.105263157894704	43.1578947368421	39.473684210526	53.1578947368421
PTTEP	57.368421052631604	57.368421052631604	52.631578947368396	55.789473684210	51.578947368421105	50.526315789473	57.368421052631604	57.894736842105296	53.1578947368421
PTTGC	53.684210526315	56.315789473684	54.2105263157895	55.263157894736	56.842105263157904	55.789473684210	54.2105263157895	56.842105263157904	53.1578947368421
ROBINS	57.368421052631604	57.368421052631604	55.789473684210	57.368421052631604	55.263157894736	52.1052631578947	57.894736842105296	57.368421052631604	55.263157894736
SAWAD	58.88888888888889	58.88888888888889	44.44444444444444	55.55555555555556	53.33333333333333	58.88888888888889	58.88888888888889	58.88888888888889	44.44444444444444
SCB	42.631578947368396	45.789473684210	53.1578947368421	49.473684210526	52.631578947368396	53.684210526315	45.263157894736	45.789473684210	51.578947368421105
SCC	48.947368421052	47.36842105263	45.789473684210	44.736842105263	46.842105263157904	44.736842105263	48.421052631578895	47.36842105263	45.789473684210
TASCO	52.631578947368396	49.4736842105263	47.368421052631	45.789473684210	45.263157894736	45.263157894736	53.157894736842	46.315789473684	47.368421052631
TCAP	46.31578947368421	44.736842105263	43.684210526315795	46.315789473684	44.210526315789	41.57894736842211	45.789473684210	44.210526315789	43.684210526315795
TMB	52.1052631578947	50.0	46.842105263157904	45.263157894736	42.631578947368396	43.684210526315795	49.473684210526	50.0	46.842105263157904
TOP	49.473684210263	47.89473684210553	50.0	48.421052631578895	45.263157894736	42.105263157894704	49.473684210263	48.421052631578895	50.0
TPIPL	47.89473684210553	48.421052631578895	44.2105263157895	50.526315789473	49.473684210263	53.15789473684211	47.89473684210553	48.421052631578895	44.2105263157895
TRUE	50.0	49.473684210263	52.631578947368396	49.473684210263	51.052631578947405	47.89473684210553	50.0	50.0	53.157894736842
TTW	43.684210526315795	43.15789473684211	44.2105263157895	44.736842105263	35.789473684210	34.736842105263	44.210526315789	43.684210526315795	44.2105263157895
TU	44.736842105263	44.736842105263	44.2105263157895	51.578947368421105	50.526315789473	48.421052631578895	46.315789473684	44.736842105263	44.736842105263
WHA	49.473684210263	48.421052631578895	49.473684210263	52.1052631578947	52.1052631578947	47.368421052631	50.0	47.89473684210553	50.0

Table 91 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	40.52631578947337	43.684210526315795	43.684210526315795	42.631578947368396	47.36842105263316	55.78947368421105	40.0	43.15789473684211	44.21052631578895
AOT	46.31578947368421	41.05263157894474	38.94736842105263	46.842105263157904	52.631578947368396	47.89473684210053	44.21052631578895	42.105263157894704	38.94736842105263
BA	37.037037037037	37.037037037037	38.888888888888889	35.185185185185205	38.888888888888889	57.4074074074074	33.333333333333	38.888888888888889	38.888888888888889
BANPU	42.631578947368396	43.15789473684211	41.57894736842211	52.1052631578947	46.842105263157904	49.473684210526	42.105263157894704	41.05263157894474	41.57894736842211
BBL	46.31578947368421	46.31578947368421	43.157894736842	45.2631578947368	46.31578947368421	51.052631578947405	44.7368421052632	46.842105263157904	44.73684210526232
BCP	41.05263157894474	41.05263157894474	38.421052631578	42.105263157894704	43.684210526315795	53.68421052631578	40.526315789473	41.57894736842211	38.42105263157895
BDMS	38.42105263157895	36.84210526315789	37.368421052631604	46.842105263157904	44.7368421052632	48.94736842105263	38.42105263157895	36.84210526315789	37.368421052631604
BEC	52.1052631578947	53.684210526315	53.157894736842	49.4736842105263	50.0	59.473684210526	52.631578947368396	53.157894736842	53.157894736842
BH	47.89473684210053	43.15789473684211	49.473684210526	42.631578947368396	47.89473684210053	46.31578947368421	48.421052631578895	44.21052631578895	49.473684210526
BLA	44.21052631578895	44.21052631578895	45.789473684210	53.157894736842	48.421052631578895	48.421052631578895	44.21052631578895	45.263157894736	45.789473684210
BTS	43.684210526315795	45.26315789473368	40.52631578947337	42.105263157894704	50.0	48.94736842105263	44.73684210526232	45.263157894736	40.52631578947337
CBG	34.0	34.0	32.0	44.0	40.0	62.0	34.0	34.0	32.0
CENTEL	38.42105263157895	39.473684210526	38.42105263157895	57.894736842105296	50.0	47.368421052631	40.52631578947337	38.42105263157895	38.42105263157895
CK	38.94736842105263	38.94736842105263	38.94736842105263	48.421052631578895	44.210526315789	50.0	40.52631578947337	38.94736842105263	38.94736842105263
CPALL	43.15789473684211	43.684210526315795	43.157894736842	44.7368421052632	46.842105263157904	53.157894736842	42.105263157894704	43.15789473684211	43.15789473684211
CPF	51.052631578947405	48.94736842105263	43.684210526315795	50.0	55.263157894736	51.578947368421105	50.52631578947337	52.631578947368396	43.684210526315795
CPN	50.0	49.473684210526	51.578947368421105	48.947368421052	48.947368421052	45.26315789473368	49.473684210526	51.578947368421105	51.578947368421105
DELTA	47.89473684210053	47.368421052631	46.315789473684	44.7368421052632	48.947368421052	46.842105263157904	47.89473684210053	47.368421052631	46.315789473684
DTAC	46.31578947368421	48.421052631578895	50.52631578947337	51.578947368421105	43.684210526315795	49.473684210526	48.421052631578895	47.89473684210053	50.52631578947337
EGCO	38.94736842105263	38.94736842105263	38.94736842105263	42.105263157894704	41.052631578974	46.315789473684	38.421052631578	38.94736842105263	38.94736842105263
GLOW	41.57894736842211	46.31578947368421	42.631578947368396	46.842105263157904	47.89473684210053	53.684210526315	45.789473684210	42.631578947368396	42.631578947368396
GPSC	43.75	43.75	43.75	56.25	56.25	62.5	43.75	43.75	43.75
HMPRO	36.84210526315789	36.84210526315789	36.84210526315789	41.05263157894474	43.684210526315795	52.631578947368396	37.368421052631604	37.894736842105296	36.84210526315789
INTUCH	40.52631578947337	41.57894736842211	42.105263157894704	50.52631578947337	44.210526315789	49.473684210263	40.52631578947337	41.57894736842211	42.105263157894704
IRPC	44.7368421052632	47.89473684210053	43.15789473684211	53.157894736842	50.0	51.578947368421105	45.789473684210	46.31578947368421	43.15789473684211
IVL	45.26315789473368	43.15789473684211	40.52631578947337	49.473684210263	42.105263157894704	41.05263157894474	45.263157894736	43.684210526315795	40.52631578947337
KBANK	46.31578947368421	48.421052631578895	45.26315789473368	52.631578947368396	50.52631578947337	54.210526315895	47.36842105263316	45.789473684210	45.26315789473368
KCE	40.52631578947337	41.05263157894474	40.52631578947337	44.7368421052632	44.7368421052632	43.684210526315795	40.0	40.52631578947337	40.52631578947337
KTB	40.52631578947337	40.52631578947337	39.473684210263	43.157894736842	41.578947368421	47.36842105263316	41.052631578974	39.473684210263	39.473684210263
LH	38.42105263157895	39.473684210263	38.42105263157895	55.263157894736 8	55.263157894736 8	48.421052631578895	41.052631578974	39.473684210263	38.42105263157895
MINT	43.684210526315795	42.631578947368396	42.631578947368396	52.631578947368396	50.52631578947337	48.421052631578895	53.157894736842	42.631578947368396	42.631578947368396
MTLS	51.020408163265294	51.020408163265294	48.9795918367347	46.938775102041	48.9795918367347	51.020408163265294	48.9795918367347	48.9795918367347	48.9795918367347
PS	41.05263157894474	40.52631578947337	36.315789473684 2	52.157894736842	49.473684210263	50.0	42.631578947368396	41.052631578974	36.315789473684 2
PTT	39.473684210263	40.52631578947337	41.05263157894474	47.36842105263316	51.052631578947405	46.315789473684 2	38.42105263157895	40.0	41.05263157894474
PTTEP	51.578947368421105	53.157894736842	54.210526315895	54.210526315895	49.473684210263	50.526315789473	50.0	53.157894736842	54.210526315895
PTTGC	48.421052631578895	49.473684210263	50.52631578947337	52.1052631578947	53.157894736842	51.578947368421105	50.0	50.0	50.52631578947337
ROBINS	54.210526315895	54.210526315895	49.473684210263	55.789473684210 5	53.684210526315 8	49.473684210263	49.473684210526	53.157894736842	50.0
SAWAD	50.0	51.111111111111109	41.111111111111	63.333333333333	55.555555555556	48.888888888889	50.0	47.777777777778	41.111111111111
SCB	45.789473684210 5	45.789473684210 5	42.105263157894704	47.36842105263316	48.421052631578895	48.421052631578895	47.368421052631 6	44.210526315895	42.105263157894704
SCC	42.105263157894704	38.42105263157895	40.0	48.947368421052 6	48.421052631578895	50.0	38.947368421052 6	38.42105263157895	40.0
TASCO	45.263157894736 8	45.789473684210 5	45.789473684210 5	52.631578947368396	49.473684210263	50.0	45.263157894736 8	46.315789473684 2	45.789473684210 5
TCAP	38.42105263157895	38.947368421052 6	38.42105263157895	47.36842105263316	47.894736842105 3	48.947368421052 6	38.42105263157895	38.42105263157895	38.42105263157895
TMB	43.684210526315795	43.684210526315795	43.684210526315795	48.947368421052 6	47.894736842105 3	50.0	43.684210526315795	43.684210526315795	43.684210526315795
TOP	41.57894736842211	41.05263157894474	40.0	43.684210526315795	48.421052631578895	44.736842105263 2	43.157894736842	42.105263157894704	40.0
TPIPL	46.315789473684 2	45.263157894736 8	44.210526315895	42.631578947368396	43.684210526315795	47.36842105263316	45.263157894736 8	44.210526315895	44.210526315895
TRUE	44.736842105263 2	46.842105263157904	45.789473684210 5	45.789473684210 5	51.052631578947405	53.157894736842 1	43.684210526315795	45.789473684210 5	45.263157894736 8
TTW	34.736842105263 2	35.789473684210 5	34.736842105263 2	46.315789473684 2	45.789473684210 5	51.052631578947405	34.736842105263 2	34.736842105263 2	34.736842105263 2
TU	41.57894736842211	41.57894736842211	40.0	52.631578947368396	45.789473684210 5	47.89473684210053	41.578947368421 1	40.526315789473 7	40.0
WHA	55.263157894736 8	54.736842105263 19	44.736842105263 2	55.263157894736 8	44.736842105263 2	48.947368421052 6	54.210526315895	54.736842105263 19	44.736842105263 2

PredictDirection11 : ATR (Average True Range)

Table 92 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.21052631578895	42.631578947368396	44.7368421052632	43.684210526315795	43.157894736842 1	43.684210526315795	43.684210526315795	42.631578947368396	44.7368421052632
AOT	60.0	60.0	61.052631578947405	60.0	60.0	61.578947368421 1	60.0	60.0	61.052631578947405
BA	37.037037037037	46.296296296305	53.703703703036695	40.740740740740 7	42.592592592595	64.814814814814 81	37.037037037037	42.592592592595	53.703703703036695
BANPU	42.105263157894704	42.631578947368396	43.157894736842 1	41.578947368421 1	42.631578947368396	41.578947368421 1	42.105263157894704	43.157894736842 1	43.157894736842 1
BBL	40.52631578947337	39.473684210263	42.105263157894704	41.578947368421 1	40.0	42.105263157894704	40.52631578947337	41.578947368421 1	42.105263157894704
BCP	51.052631578947405	50.52631578947337	56.315789473684 2	48.421052631578895	50.52631578947337	52.1052631578947	50.0	50.0	56.315789473684 2
BDMS	61.578947368421 1	63.157894736842 1	61.578947368421 1	62.105263157894 7	62.105263157894 7	62.631578947368 4	61.578947368421 1	63.157894736842 1	60.52631578947369
BEC	45.789473684210 5	46.315789473684 2	45.263157894736 8	45.263157894736 8	45.789473684210 5	45.789473684210 5	45.789473684210 5	46.315789473684 2	45.263157894736 8
BH	50.52631578947337	50.52631578947337	50.0	48.947368421052 6	52.631578947368396	49.473684210263	51.052631578947405	50.52631578947337	50.52631578947337
BLA	47.36842105263316	46.315789473684 2	51.052631578947405	46.315789473684 2	45.789473684210 5	50.52631578947337	48.421052631578895	46.315789473684 2	51.052631578947405
BTS	52.1052631578947	53.157894736842 1	56.315789473684 2	53.157894736842 1	54.736842105263 19	56.315789473684 2	51.052631578947405	54.210526315895	56.315789473684 2
CBG	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0
CENTEL	55.263157894736 8	55.789473684210 5	55.263157894736 8	56.842105263157904	55.263157894736 8	55.789473684210 5	54.210526315895	55.789473684210 5	55.263157894736 8
CK	51.052631578947405	54.210526315895	57.368421052631604	52.631578947368396	55.263157894736 8	57.368421052631604	51.052631578947405	54.210526315895	55.789473684210 5
CPALL	55.789473684210 5	54.736842105263 19	55.789473684210 5	55.263157894736 8	54.210526315895	52.631578947368396	55.789473684210 5	55.789473684210 5	55.263157894736 8
CPF	50.0	46.842105263157904	50.0	48.947368421052 6	48.421052631578895	50.0	49.473684210263	46.842105263157904	50.0
CPN	57.894736842105296	59.473684210263	60.0	58.421052631578995	58.421052631578995	56.315789473684 2	57.894736842105296	60.0	60.0
DELTA	53.684210526315 8	55.263157894736 8	58.947368421052 6	55.263157894736 8	54.210526315895	57.894736842105296	54.210526315895	55.263157894736 8	58.947368421052 6
DTAC	50.0	50.526315789473 7	48.947368421052 6	50.0	50.526315789473 7	48.947368421052 6	50.0	50.0	48.947368421052 6
EGCO	54.736842105263 19	60.526315789473 69	57.894736842105296	58.947368421052 6	60.0	56.842105263157904	55.789473684210 5	60.0	58.421052631578995
GLOW	47.36842105263316	56.315789473684 2	56.842105263157904	49.473684210263	56.315789473684 2	57.894736842105296	47.894736842105 3	54.736842105263 19	56.842105263157904
GPSC	56.25	56.25	50.0	56.25	56.25	43.75	43.75	56.25	50.0
HMPRO	58.421052631578995	56.315789473684 2	57.894736842105296	56.842105263157904	57.368421052631604	56.263157894736 8	57.368421052631604	56.842105263157904	57.894736842105296
INTUCH	40.52631578947337	41.57894736842211	41.57894736842211	40.526315789473 7	41.052631578974	42.105263157894704	42.105263157894704	41.052631578974	41.57894736842211
IRPC	56.842105263157904	56.263157894736 8	57.368421052631604	57.894736842105296	54.210526315895	56.263157894736 8	57.368421052631604	56.315789473684 2	56.842105263157904
IVL	54.210526315895	54.210526315895	53.157894736842 1	55.789473684210 5	55.789473684210 5	53.157894736842 1	56.842105263157904	54.210526315895	53.157894736842 1
KBANK	48.947368421052 6	48.421052631578895	46.315789473684 2	48.421052631578895	47.894736842105 3	44.736842105263 2	49.473684210263	48.421052631578895	46.315789473684 2
KCE	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
KTB	47.89473684210053	45.263157894736 8	47.894736842105 3	47.894736842105 3	44.736842105263 2	45.263157894736 8	47.894736842105 3	44.736842105263 2	48.421052631578895
LH	49.473684210263	54.736842105263 19	55.789473684210 5	53.157894736842 1	54.736842105263 19	58.421052631578995	48.947368421052 6	54.736842105263 19	56.315789473684 2
MINT	57.894736842105296	57.894736842105296	57.894736842105296	57.368421052631604	57.368421052631604	57.894736842105296	57.894736842105296	57.368421052631604	57.894736842105296
MTLS	55.102040816326 5	59.183673469387 8	51.020408163265294	55.102040816326 5	53.061224489759	53.061224489759	55.102040816326 5	59.183673469387 8	51.020408163265294
PS	54.736842105263 19	48.421052631578895	58.421052631578995	55.789473684210 5	49.473684210263	61.578947368421 1	53.684210526315 8	44.736842105263 2	58.421052631578995
PTT	44.21052631578895	45.789473684210 5	45.789473684210 5	47.894736842105 3	44.736842105263 2	46.315789473684 2	46.842105263157904	47.894736842105 3	45.789473684210 5
PTTEP	53.684210526315 8	47.368421052631 6	47.368421052631 6	48.947368421052 6	48.947368421052 6	47.368421052631 6	54.736842105263 19	47.368421052631 6	47.368421052631 6
PTTGC	55.789473684210 5	52.1052631578947	54.210526315895	54.210526315895	53.684210526315 8	54.736842105263 19	56.842105263157904	53.157894736842 1	54.210526315895
ROBINS	57.894736842105296	57.368421052631604	58.421052631578995	58.421052631578995	58.421052631578995	54.736842105263 19	55.263157894736 8	58.421052631578995	58.421052631578995
SAWAD	47.77777777778	50.0	48.888888888889	48.888888888889	47.77777777778	51.111111111109	47.77777777778	47.77777777778	48.888888888889
SCB	42.631578947368396	43.157894736842 1	45.263157894736 8	44.736842105263 2	42.631578947368396	44.736842105263 2	43.684210526315795	43.157894736842 1	45.263157894736 8
SCC	45.263157894736 8	48.947368421052 6	51.578947368421105	48.947368421052 6	48.421052631578895	51.578947368421105	46.842105263157904	48.947368421052 6	51.578947368421105
TASCO	53.684210526315 8	53.684210526315 8	52.105263157894 7	53.684210526315 8	52.105263157894 7	53.684210526315 8	52.105263157894 7	53.157894736842 1	53.684210526315 8
TCAP	56.842105263157904	56.842105263157904	56.315789473684 2	56.842105263157904	56.842105263157904	56.842105263157904	57.894736842105296	56.842105263157904	56.842105263157904
TMB	51.578947368421105	49.473684210263	52.631578947368396	53.684210526315 8	50.526315789473 7	53.157894736842 1	53.157894736842 1	50.526315789473 7	53.157894736842 1
TOP	60.526315789473 69	60.0	59.473684210263	60.526315789473 69	59.473684210263	59.473684210263	60.0	60.0	59.473684210263
TPIPL	51.578947368421105	53.684210526315 8	60.0	50.0	50.0	64.210526315895	52.105263157894 7	53.157894736842 1	58.947368421052 6
TRUE	43.684210526315795	45.789473684210 5	51.578947368421105	44.210526315895	44.210526315895	50.0	45.263157894736 8	45.789473684210 5	51.578947368421105
TTW	54.736842105263 19	52.105263157894 7	58.947368421052 6	46.842105263157904	52.631578947368396	59.473684210263	52.105263157894 7	50.526315789473 7	58.947368421052 6
TU	53.684210526315 8	52.105263157894 7	55.263157894736 8	53.684210526315 8	51.578947368421105	54.210526315895	52.631578947368396	51.052631578947405	56.315789473684 2
WHA	54.736842105263 19	55.789473684210 5	57.368421052631604	55.263157894736 8	56.315789473684 2	56.315789473684 2	56.315789473684 2	55.789473684210 5	56.842105263157904

Table 93 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	47.8947368421053	43.684210526315795	44.73684210526232	50.0	49.47368421052632	42.631578947368396	48.9473684210526	45.78947368421052	45.26315789473684
AOT	46.31578947368421	43.684210526315795	40.0	43.15789473684211	43.684210526315795	45.26315789473684	46.842105263157904	43.15789473684211	40.0
BA	40.7407407407407	40.7407407407407	38.88888888888889	42.592592592592595	38.88888888888889	40.7407407407407	40.7407407407407	40.7407407407407	38.88888888888889
BANPU	45.78947368421052	47.8947368421053	57.368421052631604	51.578947368421105	50.526315789473684	44.736842105263232	52.105263157894737	47.36842105263116	57.368421052631604
BBL	48.421052631578895	51.052631578947405	52.1052631578947	52.631578947368396	57.894736842105296	53.68421052631158	48.9473684210526	49.473684210526263	52.1052631578947
BCP	48.421052631578895	51.578947368421105	48.421052631578895	45.78947368421052	50.526315789473684	38.94736842105526	47.8947368421053	53.157894736842105	48.947368421052526
BDMS	53.68421052631158	47.36842105263116	46.31578947368421	48.421052631578895	47.36842105263116	39.473684210526263	52.1052631578947	48.421052631578895	45.26315789473684
BEC	43.684210526315795	52.631578947368396	51.578947368421105	58.94736842105526	52.631578947368396	52.1052631578947	43.684210526315795	53.157894736842105	50.526315789473684
BH	46.842105263157904	50.526315789473684	45.78947368421052	50.0	54.736842105263119	48.421052631578895	48.421052631578895	52.1052631578947	46.31578947368421
BLA	46.31578947368421	50.526315789473684	46.842105263157904	54.736842105263119	57.894736842105296	55.263157894736842	47.8947368421053	50.526315789473684	46.842105263157904
BTS	53.157894736842105	51.578947368421105	48.421052631578895	43.157894736842105	42.631578947368396	42.631578947368396	53.157894736842105	53.157894736842105	48.421052631578895
CBG	42.0	36.0	44.0	68.0	68.0	66.0	40.0	36.0	46.0
CENTEL	46.842105263157904	44.2105263157895	44.2105263157895	47.36842105263116	45.78947368421052	42.631578947368396	47.36842105263116	41.57894736842111	45.26315789473684
CK	48.94736842105526	47.8947368421053	49.473684210526263	49.473684210526263	47.894736842105263	47.894736842105263	48.421052631578895	48.421052631578895	49.473684210526263
CPALL	51.052631578947405	44.736842105263232	46.31578947368421	45.78947368421052	47.894736842105263	41.052631578947405	52.105263157894737	45.263157894736842	46.31578947368421
CPF	51.578947368421105	54.2105263157895	54.736842105263119	51.052631578947405	52.1052631578947	52.631578947368396	52.105263157894737	53.684210526315158	54.2105263157895
CPN	45.78947368421052	46.842105263157904	51.052631578947405	46.31578947368421	51.578947368421105	38.421052631579	45.263157894736842	46.842105263157904	51.052631578947405
DELTA	45.263157894736842	48.421052631578895	40.0	50.0	51.052631578947405	47.368421052631616	46.31578947368421	48.421052631578895	40.0
DTAC	52.631578947368396	49.473684210526263	48.947368421052526	50.0	50.0	53.157894736842105	50.0	49.473684210526263	48.421052631578895
EGCO	49.473684210526263	54.736842105263119	46.842105263157904	48.947368421052526	49.473684210526263	49.473684210526263	49.473684210526263	55.263157894736842	46.842105263157904
GLOW	51.052631578947405	55.263157894736842	42.105263157894704	50.526315789473684	55.263157894736842	43.684210526315795	47.894736842105263	54.736842105263119	42.631578947368396
GPSC	50.0	43.75	43.75	43.75	43.75	43.75	43.75	43.75	43.75
HMPRO	52.1052631578947	51.578947368421105	47.894736842105263	55.789473684210526	52.1052631578947	49.473684210526263	53.684210526315158	53.157894736842105	47.894736842105263
INTUCH	52.631578947368396	48.421052631578895	43.684210526315795	51.052631578947405	50.526315789473684	46.315789473684232	52.1052631578947	46.842105263157904	43.157894736842105
IRPC	43.157894736842105	48.421052631578895	45.789473684210526	52.1052631578947	45.789473684210526	51.578947368421105	43.157894736842105	47.894736842105263	45.263157894736842
IVL	55.263157894736842	53.157894736842105	56.842105263157904	46.842105263157904	47.368421052631616	50.0	55.789473684210526	46.315789473684232	56.315789473684232
KBANK	56.842105263157904	52.1052631578947	45.789473684210526	48.421052631578895	48.421052631578895	46.315789473684232	54.2105263157895	53.157894736842105	45.789473684210526
KCE	51.578947368421105	50.526315789473684	50.0	44.736842105263232	47.368421052631616	46.315789473684232	50.526315789473684	49.473684210526263	50.0
KTB	52.1052631578947	47.368421052631616	39.473684210526263	50.526315789473684	51.052631578947405	51.578947368421105	53.684210526263158	49.473684210526263	38.947368421052526
LH	54.2105263157895	53.157894736842105	52.1052631578947	50.526315789473684	48.947368421052526	47.368421052631616	55.263157894736842	52.631578947368396	52.1052631578947
MINT	45.263157894736842	42.105263157894704	50.0	50.526315789473684	44.736842105263232	55.263157894736842	44.736842105263232	44.210526315789475	50.0
MTLS	51.020408163265294	48.97959183673347	44.89759183673535	46.938775510204145	53.061224489795959	44.89795918367735	51.020408163265294	46.938775510204145	46.938775510204145
PS	43.157894736842105	45.263157894736842	45.789473684210526	44.736842105263232	45.263157894736842	34.736842105263232	46.842105263157904	44.736842105263232	46.315789473684232
PTT	54.736842105263119	52.631578947368396	56.842105263157904	46.842105263157904	51.052631578947405	40.0	53.157894736842105	52.631578947368396	56.842105263157904
PTTEP	51.578947368421105	48.421052631578895	51.578947368421105	56.315789473684232	55.263157894736842	52.105263157895	54.210526315789475	50.526315789473684	51.578947368421105
PTTGC	53.157894736842105	52.105263157894737	52.105263157894737	54.210526315789475	52.631578947368396	45.789473684210526	53.157894736842105	52.631578947368396	52.105263157894737
ROBINS	60.0	58.421052631578995	61.578947368421105	48.947368421052526	46.842105263157904	45.263157894736842	54.736842105263119	58.421052631578995	62.105263157894737
SAWAD	56.666666666667	48.888888888889	56.666666666667	44.444444444444	50.0	41.111111111111	57.777777777779	48.888888888889	56.666666666667
SCB	50.526315789473684	55.263157894736842	58.421052631578995	48.421052631578895	51.052631578947405	51.052631578947405	52.631578947368396	56.315789473684232	57.894736842105296
SCC	44.210526315789475	51.052631578947405	48.421052631578895	50.0	47.894736842105263	44.210526315789475	45.263157894736842	50.0	49.473684210526263
TASCO	47.894736842105263	46.315789473684232	45.263157894736842	45.789473684210526	44.736842105263232	46.315789473684232	46.315789473684232	46.315789473684232	45.263157894736842
TCAP	44.736842105263232	45.263157894736842	46.315789473684232	46.842105263157904	43.157894736842105	53.157894736842105	47.894736842105263	45.263157894736842	46.315789473684232
TMB	46.842105263157904	49.473684210526263	49.473684210526263	49.473684210526263	52.631578947368396	47.894736842105263	47.368421052631616	50.0	48.947368421052526
TOP	47.894736842105263	47.894736842105263	43.684210526315795	50.0	53.157894736842105	44.210526315789475	46.842105263157904	47.368421052631616	44.210526315789475
TPIPL	40.0	47.368421052631616	44.73684210526232	52.631578947368396	48.421052631578895	44.210526315789475	39.473684210526263	46.842105263157904	45.263157894736842
TRUE	50.526315789473684	53.684210526263158	45.263157894736842	49.473684210526263	52.631578947368396	44.736842105263232	50.526315789473684	52.631578947368396	45.263157894736842
TTW	44.736842105263232	42.631578947368396	39.473684210526263	38.947368421052526	41.052631578947474	39.473684210526263	44.736842105263232	42.631578947368396	39.473684210526263
TU	48.947368421052526	46.842105263157904	52.105263157894737	44.210526315789475	45.789473684210526	40.0	48.947368421052526	47.894736842105263	52.105263157894737
WHA	45.789473684210526	48.947368421052526	56.315789473684232	48.421052631578895	51.052631578947405	46.315789473684232	44.736842105263232	48.947368421052526	56.315789473684232

Table 94 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	45.26315789473684	47.36842105263116	45.78947368421052	44.736842105263232	45.263157894736842	45.78947368421052	46.842105263157904	47.894736842105263	46.31578947368421
AOT	51.052631578947405	51.052631578947405	44.210526315789475	41.052631578947474	42.105263157894704	41.052631578947474	52.1052631578947	51.052631578947405	44.210526315789475
BA	40.7407407407407	38.88888888888889	38.88888888888889	46.296296296296305	44.444444444444	44.444444444444	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	45.78947368421052	46.842105263157904	57.894736842105296	48.947368421052526	51.052631578947405	44.210526315789475	50.526315789473684	47.368421052631616	57.894736842105296
BBL	46.31578947368421	53.684210526263158	51.578947368421105	53.157894736842105	52.631578947368396	49.473684210526263	46.842105263157904	54.210526315789475	52.105263157894737
BCP	48.421052631578895	50.0	48.421052631578895	42.105263157894704	50.0	38.947368421052526	46.842105263157904	50.0	48.947368421052526
BDMS	51.578947368421105	47.894736842105263	45.789473684210526	51.578947368421105	47.368421052631616	39.473684210526263	52.631578947368396	48.947368421052526	45.263157894736842
BEC	45.789473684210526	48.421052631578895	52.105263157894737	58.947368421052526	53.157894736842105	52.105263157894737	46.842105263157904	49.473684210526263	51.578947368421105
BH	51.052631578947405	52.631578947368396	46.315789473684232	55.263157894736842	55.263157894736842	48.947368421052526	51.052631578947405	51.578947368421105	46.315789473684232
BLA	48.421052631578895	51.052631578947405	46.842105263157904	55.263157894736842	56.842105263157904	55.263157894736842	46.842105263157904	51.052631578947405	46.842105263157904
BTS	53.157894736842105	54.736842105263119	50.526315789473737	46.842105263157904	43.684210526315795	41.578947368421105	54.210526315789475	53.684210526263158	50.526315789473737
CBG	32.0	36.0	44.0	68.0	44.0	66.0	38.0	36.0	46.0
CENTEL	46.31578947368421	44.2105263157895	47.36842105263116	46.842105263157904	47.894736842105263	40.0	48.421052631578895	44.2105263157895	46.842105263157904
CK	44.210526315789475	48.947368421052526	47.894736842105263	44.736842105263232	46.842105263157904	51.578947368421105	44.210526315789475	46.315789473684232	48.421052631578895
CPALL	49.473684210526263	45.263157894736842	46.315789473684232	46.315789473684232	47.894736842105263	41.052631578947474	50.526315789473737	44.736842105263232	46.315789473684232
CPF	51.052631578947405	53.684210526263158	53.157894736842105	54.210526315789475	50.526315789473737	52.631578947368396	50.526315789473737	53.684210526263158	53.157894736842105
CPN	46.842105263157904	45.789473684210526	51.052631578947405	43.684210526315795	51.052631578947405	38.421052631579	46.315789473684232	45.789473684210526	51.052631578947405
DELTA	45.263157894736842	48.421052631578895	40.0	50.526315789473737	50.526315789473737	47.368421052631616	46.842105263157904	47.894736842105263	40.0
DTAC	52.1052631578947	50.0	51.578947368421105	48.421052631578895	49.473684210526263	52.105263157894737	52.631578947368396	52.1052631578947	52.631578947368396
EGCO	48.947368421052526	53.157894736842105	46.842105263157904	47.894736842105263	49.473684210526263	49.473684210526263	50.526315789473737	54.736842105263119	46.842105263157904
GLOW	51.578947368421105	54.210526315789475	42.631578947368396	50.526315789473737	55.263157894736842	43.684210526315795	51.052631578947405	55.263157894736842	42.631578947368396
GPSC	50.0	43.75	43.75	43.75	43.75	43.75	43.75	43.75	43.75
HMPRO	52.631578947368396	55.263157894736842	50.0	56.315789473684232	56.842105263157904	50.526315789473737	52.631578947368396	52.1052631578947	50.0
INTUCH	56.315789473684232	47.894736842105263	43.684210526315795	54.736842105263119	49.473684210526263	46.315789473684232	54.2105263157895	47.368421052631616	43.157894736842105
IRPC	44.736842105263232	48.947368421052526	45.789473684210526	45.789473684210526	45.789473684210526	47.894736842105263	45.263157894736842	51.052631578947405	45.263157894736842
IVL	55.263157894736842	54.210526315789475	56.842105263157904	45.789473684210526	48.421052631578895	50.0	53.684210526263158	54.210526315789475	56.842105263157904
KBANK	54.736842105263119	52.1052631578947	45.263157894736842	48.421052631578895	47.894736842105263	47.368421052631616	53.684210526263158	53.157894736842105	45.263157894736842
KCE	49.473684210526263	48.421052631578895	51.578947368421105	44.736842105263232	46.842105263157904	45.263157894736842	50.526315789473737	50.0	51.578947368421105
KTB	51.578947368421105	47.368421052631616	39.473684210526263	52.1052631578947	51.052631578947405	51.578947368421105	52.1052631578947	47.368421052631616	38.947368421052526
LH	56.842105263157904	53.157894736842105	50.0	53.684210526263158	52.631578947368396	45.263157894736842	52.631578947368396	52.631578947368396	50.0
MINT	40.0	42.631578947368396	51.578947368421105	44.210526315789475	43.157894736842105	53.157894736842105	43.684210526315795	42.631578947368396	51.578947368421105
MTLS	59.1836734693878	48.97959183673347	44.89759183673535	46.938775510204145	44.897959183673535	44.897959183673535	48.97959183673347	51.020408163265294	46.938775510204145
PS	42.631578947368396	45.263157894736842	44.736842105263232	36.842105263157879	38.947368421052526	40.0	42.105263157894704	44.736842105263232	44.736842105263232
PTT	53.684210526263158	52.631578947368396	56.315789473684232	46.842105263157904	52.105263157894737	40.0	53.684210526263158	52.631578947368396	56.315789473684232
PTTEP	53.157894736842105	49.473684210526263	52.631578947368396	54.736842105263119	52.105263157894737	52.105263157894737	53.157894736842105	48.421052631578895	52.631578947368396
PTTGC	50.526315789473737	53.157894736842105	52.105263157894737	53.157894736842105	54.210526315789475	45.789473684210526	52.631578947368396	52.105263157894737	52.105263157894737
ROBINS	58.421052631578995	58.421052631578995	61.578947368421105	48.421052631578895	45.789473684210526	45.263157894736842	58.947368421052526	58.421052631578995	62.105263157894737
SAWAD	58.888888888889	48.888888888889	56.666666666667	46.666666666667	52.222222222222	41.111111111111	58.888888888889	56.666666666667	56.666666666667
SCB	52.105263157894737	56.315789473684232	58.421052631578995	50.0	53.157894736842105	51.052631578947405	52.631578947368396	56.842105263157904	57.894736842105296
SCC	47.894736842105263	48.421052631578895	48.421052631578895	46.315789473684232	42.105263157894704	42.105263157894704	47.894736842105263	48.421052631578895	48.421052631578895
TASCO	46.842105263157904	46.315789473684232	46.842105263157904	47.368421052631616	46.315789473684232	45.263157894736842	47.368421052631616	46.315789473684232	46.315789473684232
TCAP	45.789473684210526	45.263157894736842	46.315789473684232	48.421052631578895	44.210526315789475	53.157894736842105	45.789473684210526	46.315789473684232	46.315789473684232
TMB	46.315789473684232	49.473684210526263	48.947368421052526	48.421052631578895	52.105263157894737	47.368421052631616	46.842105263157904	51.052631578947405	49.473684210526263
TOP	47.894736842105263	47.894736842105263	43.684210526315795	50.526315789473737	52.631578947368396	44.210526315789475	48.947368421052526	48.421052631578895	44.210526315789475
TPIPL	40.0	42.631578947368396	37.894736842105296	50.0	44.210526315789475	44.210526315789475	40.526315789473737	45.263157894736842	37.368421052631604
TRUE	46.842105263157904	47.894736842105263	45.789473684210526	44.210526315789475	43.157894736842105	42.631578947368396	47.368421052631616	48.947368421052526	45.789473684210526
TTW	45.263157894736842	50.526315789473737	40.0	40.0	41.052631578947474	42.105263157894704	45.263157894736842	50.526315789473737	40.0
TU	48.421052631578895	47.368421052631616	52.105263157894737	46.842105263157904	46.842105263157904	40.0	44.210526315789475	47.368421052631616	52.105263157894737
WHA	43.157894736842105	46.315789473684232	51.052631578947405	52.631578947368396	51.052631578947405	49.473684210526263	44.736842105263232	47.894736842105263	51.052631578947405

Table 95 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.631578947368396	44.7368421052632	43.684210526315795	49.473684210526 3	45.26315789473 68	54.2105263157895	44.7368421052632	46.842105263157904	43.684210526315795
AOT	38.4210526315789	40.0	38.9473684210526	54.736842105263 19	50.526315789473 7	49.473684210526 3	38.4210526315789	40.0	38.4210526315789
BA	37.037037037037	38.888888888888 9	38.888888888888 9	42.592592592595	40.740740740740 7	50.0	37.037037037037	38.888888888888 9	38.888888888888 9
BANPU	45.26315789473 68	44.7368421052632	41.57894736842 11	57.368421052631 604	42.631578947368396	44.7368421052632	46.31578947368 42	47.36842105263 16	41.57894736842 11
BBL	50.0	54.2105263157895	44.7368421052632	48.421052631578895	55.26315789473 68	55.78947368421 05	53.15789473684 21	53.68421052631 58	44.7368421052632
BCP	40.526315789473 7	42.631578947368396	38.4210526315789	51.052631578947405	45.26315789473 68	53.68421052631 58	41.57894736842 11	42.105263157894704	38.4210526315789
BDMS	40.526315789473 7	41.05263157894 74	37.368421052631604	51.052631578947405	53.15789473684 21	54.736842105263 19	40.526315789473 7	41.05263157894 74	37.368421052631604
BEC	48.9473684210526	43.684210526315795	45.263157894736 8	51.052631578947405	51.052631578947405	51.578947368421105	45.78947368421 05	48.9473684210526	45.263157894736 8
BH	42.105263157894704	48.9473684210526	49.4736842105263	50.0	48.9473684210526	44.7368421052632	46.31578947368 42	48.421052631578895	49.4736842105263
BLA	45.78947368421 05	41.57894736842 11	45.78947368421 05	45.263157894736 8	46.31578947368 42	47.8947368421053	45.78947368421 05	41.05263157894 74	45.78947368421 05
BTS	43.684210526315795	40.0	40.526315789473 7	48.421052631578895	41.05263157894 74	40.0	42.631578947368396	41.05263157894 74	40.526315789473 7
CBG	32.0	34.0	32.0	32.0	34.0	46.0	32.0	34.0	32.0
CENTEL	41.05263157894 74	44.7368421052632	41.57894736842 11	56.842105263157904	48.9473684210526	44.7368421052632	42.631578947368396	43.157894736842 1	41.57894736842 11
CK	40.0	40.0	38.9473684210526	54.736842105263 19	45.78947368421 05	51.578947368421105	41.05263157894 74	40.526315789473 7	38.9473684210526
CPALL	42.105263157894704	45.78947368421 05	43.157894736842 1	52.631578947368396	52.105263157894 7	51.578947368421105	43.157894736842 1	46.842105263157904	43.157894736842 1
CPF	48.421052631578895	46.315789473684 2	43.684210526315795	47.894736842105 3	48.421052631578895	45.26315789473 68	48.9473684210526	46.842105263157904	43.684210526315795
CPN	40.0	46.315789473684 2	38.4210526315789	44.210526315789 5	42.105263157894704	44.7368421052632	44.7368421052632	43.157894736842 1	38.4210526315789
DELTA	46.842105263157904	46.842105263157904	45.263157894736 8	51.052631578947405	55.263157894736 3	53.157894736842 1	45.78947368421 05	45.78947368421 05	45.263157894736 8
DTAC	44.2105263157895	46.315789473684 2	52.631578947368396	44.736842105263 2	48.421052631578895	54.2105263157895	45.78947368421 05	46.842105263157904	53.157894736842 1
EGCO	44.2105263157895	55.789473684210 5	44.736842105263 2	42.631578947368396	57.894736842105296	44.7368421052632	44.2105263157895	56.315789473684 2	44.736842105263 2
GLOW	51.052631578947405	52.1052631578947	40.526315789473 7	50.526315789473 7	42.631578947368396	52.1052631578947	47.8947368421053	50.0	40.526315789473 7
GPSC	43.75	43.75	43.75	50.0	56.25	43.75	43.75	43.75	43.75
HMPRO	37.368421052631604	41.05263157894 74	36.842105263157 9	50.526315789473 7	49.473684210526 3	58.421052631578995	36.842105263157 9	40.0	36.842105263157 9
INTUCH	42.631578947368396	47.894736842105 3	42.105263157894704	55.263157894736 8	48.9473684210526	47.894736842105 3	42.105263157894704	46.842105263157904	42.105263157894704
IRPC	42.631578947368396	45.26315789473 68	43.157894736842 1	54.2105263157895	44.736842105263 2	51.052631578947405	43.157894736842 1	45.78947368421 05	43.157894736842 1
IVL	41.57894736842 11	42.631578947368396	42.631578947368396	45.78947368421 05	46.31578947368 42	47.894736842105 3	43.157894736842 1	42.631578947368396	42.631578947368396
KBANK	48.421052631578895	46.842105263157904	45.26315789473 68	48.9473684210526	49.473684210526 3	55.263157894736 8	48.421052631578895	48.421052631578895	45.26315789473 68
KCE	45.78947368421 05	47.368421052631 6	40.0	43.157894736842 1	47.894736842105 3	52.631578947368396	46.842105263157904	50.0	40.0
KTB	43.684210526315795	48.421052631578895	39.473684210526 3	60.526315789473 69	53.157894736842 1	48.421052631578895	44.2105263157895	45.78947368421 05	39.473684210526 3
LH	53.157894736842 1	50.526315789473 7	38.4210526315789	51.052631578947405	52.631578947368396	45.78947368421 05	52.1052631578947	51.578947368421105	38.4210526315789
MINT	42.631578947368396	42.105263157894704	42.105263157894704	43.684210526315795	46.842105263157904	47.368421052631 6	43.157894736842 1	41.57894736842 11	42.105263157894704
MTLS	44.8979591836735	42.857142857142 9	48.9795918367347	55.102040816326 5	46.938775510204 1	44.8979591836735	48.9795918367347	44.8979591836735	48.9795918367347
PS	36.315789473684 2	40.526315789473 7	36.315789473684 2	47.894736842105 3	50.0	42.105263157894704	36.315789473684 2	41.05263157894 74	36.315789473684 2
PTT	50.526315789473 7	52.631578947368396	36.315789473684 2	54.736842105263 19	48.9473684210526	58.947368421052 6	50.526315789473 7	52.631578947368396	36.315789473684 2
PTTEP	56.315789473684 2	53.684210526315 8	54.210526315795	47.894736842105 3	48.9473684210526	42.631578947368396	56.315789473684 2	58.421052631578995	54.210526315795
PTTGC	43.157894736842 1	46.842105263157904	46.842105263157904	48.421052631578895	42.631578947368396	47.368421052631 6	44.2105263157895	48.421052631578895	46.842105263157904
ROBINS	46.315789473684 2	47.894736842105 3	37.894736842105296	41.57894736842 11	48.421052631578895	47.894736842105 3	47.368421052631 6	46.315789473684 2	37.894736842105296
SAWAD	51.111111111111 09	43.333333333333 3	41.111111111111 1	55.555555555556	47.777777777778	54.444444444444 39	43.333333333333 3	45.555555555556	41.111111111111 1
SCB	50.526315789473 7	50.0	41.052631578947 4	45.78947368421 05	50.0	53.157894736842 1	48.947368421052 6	50.0	41.052631578947 4
SCC	42.105263157894704	41.05263157894 74	40.0	48.9473684210526	43.157894736842 1	50.526315789473 7	41.05263157894 74	42.105263157894704	40.0
TASCO	46.842105263157904	48.947368421052 6	46.315789473684 2	53.684210526315 8	48.9473684210526	47.368421052631 6	48.421052631578895	48.947368421052 6	46.315789473684 2
TCAP	40.526315789473 7	40.0	38.4210526315789	54.736842105263 19	45.263157894736 8	47.894736842105 3	42.105263157894704	41.05263157894 74	38.4210526315789
TMB	45.263157894736 8	44.736842105263 2	43.684210526315795	46.315789473684 2	46.315789473684 2	46.315789473684 2	45.263157894736 8	43.684210526315795	43.684210526315795
TOP	48.9473684210526	46.315789473684 2	40.0	53.157894736842 1	48.9473684210526	45.78947368421 05	47.368421052631 6	48.9473684210526	40.0
TPIPL	41.57894736842 11	43.157894736842 1	44.210526315795	46.842105263157904	50.0	59.473684210263	41.052631578947 4	48.421052631578895	44.210526315795
TRUE	45.78947368421 05	47.368421052631 6	45.263157894736 8	52.631578947368396	44.736842105263 2	45.263157894736 8	44.2105263157895	46.315789473684 2	45.263157894736 8
TTW	34.736842105263 2	35.263157894736 8	34.736842105263 2	45.263157894736 8	42.631578947368396	52.631578947368396	34.736842105263 2	35.263157894736 8	34.736842105263 2
TU	45.263157894736 8	43.157894736842 1	40.0	46.842105263157904	49.473684210526 3	54.2105263157895	43.684210526315795	43.157894736842 1	40.0
WHA	44.2105263157895	44.2105263157895	44.736842105263 2	52.631578947368396	52.631578947368396	48.421052631578895	45.263157894736 8	46.315789473684 2	44.736842105263 2

PredictDirection12 : EMD of closing price

Table 96 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.7368421052632	44.7368421052632	44.2105263157895	44.7368421052632	43.684210526315795	43.684210526315795	44.7368421052632	44.7368421052632	44.2105263157895
AOT	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405
BA	44.444444444444 4	48.148148148148 1	55.555555555556	50.0	53.703703703703695	53.703703703703695	44.444444444444 4	48.148148148148 1	55.555555555556
BANPU	41.052631578947 4	41.052631578947 4	42.631578947368396	41.052631578947 4	40.0	41.052631578947 4	41.052631578947 4	41.052631578947 4	42.631578947368396
BBL	40.526315789473 7	40.526315789473 7	41.57894736842 11	39.473684210526 3	41.052631578947 4	40.526315789473 7	40.526315789473 7	40.526315789473 7	41.57894736842 11
BCP	47.368421052631 6	46.842105263157904	48.421052631578895	46.842105263157904	47.368421052631 6	47.894736842105 3	48.421052631578895	46.842105263157904	48.947368421052 6
BDMS	62.10526315789 47	62.631578947368 4	61.052631578947405	62.10526315789 47	62.631578947368 4	61.052631578947405	62.10526315789 47	62.10526315789 47	61.052631578947405
BEC	44.2105263157895	45.263157894736 8	45.263157894736 8	44.2105263157895	45.263157894736 8	44.736842105263 2	44.2105263157895	45.263157894736 8	45.263157894736 8
BH	51.052631578947405	48.947368421052 6	48.947368421052 6	50.0	50.0	52.1052631578947	50.0	49.473684210526 3	48.947368421052 6
BLA	46.842105263157904	45.263157894736 8	50.0	46.315789473684 2	44.736842105263 2	48.947368421052 6	47.894736842105 3	45.263157894736 8	50.0
BTS	51.052631578947405	55.263157894736 8	48.421052631578895	50.0	54.2105263157895	46.842105263157904	51.052631578947405	55.789473684210 5	48.421052631578895
CBG	68.0	68.0	68.0	68.0	68.0	68.0	68.0	68.0	68.0
CENTEL	60.526315789473 69	59.473684210526 3	59.473684210526 3	59.473684210526 3	59.473684210526 3	58.421052631578995	60.526315789473 69	59.473684210526 3	59.473684210526 3
CK	53.157894736842 1	51.052631578947405	52.105263157894 7	53.684210526315 8	51.052631578947405	55.789473684210 5	53.684210526315 8	50.526315789473 7	52.105263157894 7
CPALL	53.157894736842 1	54.2105263157895	55.263157894736 8	55.263157894736 8	54.2105263157895	55.789473684210 5	55.263157894736 8	53.684210526315 8	55.263157894736 8
CPF	50.0	45.263157894736 8	51.052631578947405	48.947368421052 6	45.263157894736 8	51.578947368421105	48.421052631578895	45.263157894736 8	51.052631578947405
CPN	52.631578947368396	49.473684210526 3	51.052631578947405	53.684210526315 8	49.473684210526 3	53.684210526315 8	53.157894736842 1	49.473684210526 3	51.052631578947405
DELTA	51.052631578947405	53.157894736842 1	46.842105263157904	51.578947368421105	55.263157894736 8	45.789473684210 5	51.578947368421105	54.736842105263 19	46.842105263157904
DTAC	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
EGCO	56.842105263157904	54.736842105263 19	55.789473684210 5	56.842105263157904	55.263157894736 8	59.473684210263	56.842105263157904	54.210526315795	55.789473684210 5
GLOW	53.157894736842 1	53.684210526315 8	54.736842105263 19	52.1052631578947	54.736842105263 19	53.684210526315 8	52.631578947368396	53.684210526315 8	54.210526315795
GPSC	37.5	43.75	43.75	37.5	56.25	43.75	37.5	43.75	43.75
HMPRO	54.736842105263 19	56.315789473684 2	58.421052631578995	54.736842105263 19	56.315789473684 2	56.842105263157904	56.315789473684 2	56.315789473684 2	58.947368421052 6
INTUCH	43.157894736842 1	42.631578947368396	42.105263157894704	43.684210526315795	42.631578947368396	42.105263157894704	42.631578947368396	42.631578947368396	42.105263157894704
IRPC	53.157894736842 1	54.210526315789 5	51.578947368421105	53.684210526315 8	55.263157894736 8	52.631578947368396	53.157894736842 1	54.736842105263 19	51.578947368421105
IVL	50.0	50.526315789473 7	53.684210526315 8	49.473684210526 3	50.0	54.2105263157895	50.0	50.526315789473 7	53.684210526315 8
KBANK	44.736842105263 2	43.157894736842 1	43.157894736842 1	43.684210526315795	43.684210526315795	44.2105263157895	44.736842105263 2	43.157894736842 1	43.157894736842 1
KCE	58.947368421052 6	59.473684210526 3	59.473684210526 3	58.421052631578995	59.473684210526 3	59.473684210526 3	58.421052631578995	59.473684210526 3	59.473684210526 3
KTB	38.947368421052 6	39.473684210526 3	40.0	38.947368421052 6	38.947368421052 6	37.894736842105296	38.947368421052 6	39.473684210526 3	40.0
LH	52.1052631578947	50.526315789473 7	49.473684210526 3	53.157894736842 1	45.263157894736 8	52.631578947368396	50.526315789473 7	50.526315789473 7	49.473684210526 3
MINT	57.894736842105296	58.947368421052 6	57.894736842105296	57.368421052631604	58.947368421052 6	58.421052631578995	57.894736842105296	58.947368421052 6	57.894736842105296
MTLS	51.020408163265294	48.979591836734 7	53.061224489759 5	46.938775510204 1	48.979591836734 7	46.938775510204 1	51.020408163265294	48.979591836734 7	53.061224489759 5
PS	50.526315789473 7	42.631578947368396	46.315789473684 2	47.368421052631 6	44.736842105263 2	41.052631578947 4	51.578947368421105	42.631578947368396	45.789473684210 5
PTT	42.105263157894704	41.05263157894 74	51.578947368421105	43.157894736842 1	41.052631578947 4	52.631578947368396	42.105263157894704	41.05263157894 74	51.578947368421105
PTTEP	47.894736842105 3	47.368421052631 6	48.421052631578895	46.315789473684 2	47.368421052631 6	47.368421052631 6	48.421052631578895	47.368421052631 6	48.421052631578895
PTTGC	52.631578947368396	51.578947368421105	47.368421052631 6	53.684210526315 8	51.578947368421105	47.368421052631 6	51.578947368421105	52.1052631578947	47.368421052631 6
ROBINS	54.736842105263 19	54.210526315795	54.736842105263 19	54.736842105263 19	53.157894736842 1	54.736842105263 19	54.736842105263 19	53.684210526315 8	54.736842105263 19
SAWAD	41.111111111111 1	40.0	43.333333333333 3	40.0	43.333333333333 3	40.0	40.0	44.444444444444 4	43.333333333333 3
SCB	41.57894736842 11	43.157894736842 1	44.210526315795	41.57894736842 11	47.368421052631 6	42.105263157894704	43.157894736842 1	44.210526315795	
SCC	45.263157894736 8	47.894736842105 3	42.631578947368396	47.368421052631 6	47.894736842105 3	42.105263157894704	45.263157894736 8	47.894736842105 3	42.631578947368396
TASCO	48.947368421052 6	41.57894736842 11	45.263157894736 8	43.684210526315795	42.631578947368396	45.789473684210 5	49.473684210263	43.684210526315795	45.789473684210 5
TCAP	53.684210526315 8	55.789473684210 5	55.263157894736 8	54.736842105263 19	55.789473684210 5	54.2105263157895	53.684210526315 8	55.789473684210 5	55.263157894736 8
TMB	44.736842105263 2	43.684210526315795	46.315789473684 2	42.631578947368396	43.684210526315795	47.368421052631 6	44.210526315895	44.736842105263 2	46.315789473684 2
TOP	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
TPIPL	49.473684210263	56.842105263157904	50.526315789473 7	49.473684210263	53.157894736842 1	50.526315789473 7	50.0	55.263157894736 8	51.578947368421105
TRUE	55.789473684210 5	53.157894736842 1	55.263157894736 8	54.736842105263 19	55.263157894736 8	55.789473684210 5	55.789473684210 5	53.157894736842 1	55.263157894736 8
TTW	48.947368421052 6	47.368421052631 6	53.684210526315 8	48.421052631578895	47.368421052631 6	47.368421052631 6	48.421052631578895	47.368421052631 6	54.736842105263 19
TU	56.842105263157904	57.894736842105296	50.526315789473 7	54.736842105263 19	56.842105263157904	50.526315789473 7	56.842105263157904	57.894736842105296	50.526315789473 7
WHA	48.947368421052 6	46.842105263157904	47.894736842105 3	48.947368421052 6	46.315789473684 2	51.578947368421105	48.947368421052 6	46.842105263157904	47.894736842105 3

Table 97 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	50.5263157894737	44.2105263157895	43.684210526315795	50.5263157894737	43.684210526315795	43.684210526315795	50.5263157894737
AOT	48.421052631578895	51.052631578947405	50.0	51.578947368421105	51.578947368421105	46.842105263157904	48.421052631578895	51.052631578947405	50.0
BA	42.592592592592595	37.037037037037	46.296296296296305	42.592592592592595	40.7407407407407	44.4444444444444	44.4444444444444	38.88888888888889	46.296296296296305
BANPU	58.421052631578995	58.421052631578995	52.631578947368396	58.421052631578995	58.421052631578995	56.842105263157904	58.421052631578995	58.421052631578995	52.631578947368396
BBL	47.8947368421053	49.4736842105263	44.7368421052632	48.421052631578895	47.8947368421053	47.8947368421053	47.8947368421053	49.4736842105263	44.7368421052632
BCP	50.0	50.5263157894737	55.2631578947368	48.421052631578895	48.947368421052 6	52.1052631578947	50.0	50.5263157894737	55.2631578947368
BDMS	54.736842105263 19	60.0	54.2105263157895	51.578947368421105	58.421052631578995	56.842105263157904	54.736842105263 19	58.9473684210526	54.2105263157895
BEC	56.31578947368 42	55.263157894736 8	57.894736842105296	56.31578947368 42	55.78947368421 05	57.894736842105296	56.31578947368 42	55.263157894736 8	57.894736842105296
BH	46.31578947368 42	44.2105263157895	48.9473684210526	49.4736842105263	45.263157894736 8	49.4736842105263	46.31578947368 42	44.2105263157895	48.9473684210526
BLA	50.5263157894737	51.052631578947405	50.0	51.052631578947405	51.052631578947405	50.0	50.5263157894737	50.5263157894737	50.0
BTS	58.9473684210526	59.4736842105263	61.052631578947405	59.4736842105263	57.894736842105296	58.421052631578995	58.9473684210526	59.4736842105263	61.052631578947405
CBG	50.0	62.0	57.99999999999999	68.0	68.0	62.0	56.000000000000 01	60.0	54.0
CENTEL	57.894736842105296	58.9473684210526	55.2631578947368	58.421052631578995	58.421052631578995	56.842105263157904	56.842105263157904	58.9473684210526	55.2631578947368
CK	53.1578947368421	55.7894736842105	52.1052631578947	56.31578947368 42	55.78947368421 05	55.7894736842105	53.6842105263158	55.7894736842105	52.1052631578947
CPALL	53.684210526315 8	52.631578947368396	60.0	51.052631578947405	48.421052631578895	51.578947368421105	54.2105263157895	53.1578947368421	58.9473684210526
CPF	53.1578947368421	51.578947368421105	48.9473684210526	51.052631578947405	52.631578947368396	48.9473684210526	51.052631578947405	52.1052631578947	48.9473684210526
CPN	50.5263157894737	53.684210526315 8	56.842105263157904	55.7894736842105	54.7368421052619	51.578947368421105	51.578947368421105	53.684210526315 8	56.842105263157904
DELTA	54.736842105263 19	54.2105263157895	54.736842105263 19	55.2631578947368	53.684210526315 8	54.2105263157895	54.2105263157895	54.2105263157895	54.736842105263 19
DTAC	48.421052631578895	48.9473684210526	53.157894736842 1	48.421052631578895	48.9473684210526	51.052631578947405	48.421052631578895	48.9473684210526	53.157894736842 1
EGCO	50.5263157894737	47.8947368421053	53.157894736842 1	51.052631578947405	47.8947368421053	54.736842105263 19	49.4736842105263	49.4736842105263	54.736842105263 19
GLOW	46.842105263157904	46.315789473684 2	48.9473684210526	52.1052631578947	47.8947368421053	50.5263157894737	45.7894736842105	45.7894736842105	48.421052631578895
GPSC	43.75	43.75	43.75	43.75	43.75	43.75	43.75	43.75	43.75
HMPRO	56.842105263157904	58.9473684210526	47.368421052631 6	58.9473684210526	59.4736842105263	50.0	56.842105263157904	58.9473684210526	47.368421052631 6
INTUCH	45.263157894736 8	45.263157894736 8	52.1052631578947	44.7368421052632	47.8947368421053	50.5263157894737	44.7368421052632	45.263157894736 8	52.1052631578947
IRPC	54.736842105263 19	54.736842105263 19	55.7894736842105	54.2105263157895	54.736842105263 19	52.1052631578947	53.6842105263158	54.2105263157895	55.7894736842105
IVL	55.263157894736 8	51.578947368421105	53.6842105263158	51.052631578947405	51.578947368421105	53.157894736842 1	54.736842105263 19	50.0	53.6842105263158
KBANK	44.736842105263 2	45.263157894736 8	44.736842105263 2	42.631578947368396	43.684210526315795	43.684210526315795	44.736842105263 2	45.263157894736 8	44.736842105263 2
KCE	48.9473684210526	45.263157894736 8	54.736842105263 19	47.8947368421053	46.315789473684 2	49.4736842105263	50.5263157894737	46.315789473684 2	54.736842105263 19
KTB	47.368421052631 6	46.842105263157904	50.0	47.8947368421053	46.842105263157904	48.9473684210526	48.421052631578895	46.842105263157904	50.0
LH	62.6315789473684	61.57894736842 11	59.4736842105263	60.0	59.4736842105263	58.9473684210526	62.6315789473684	61.57894736842 11	59.4736842105263
MINT	54.736842105263 19	54.736842105263 19	52.631578947368396	52.631578947368396	53.157894736842 1	48.421052631578895	52.1052631578947	54.736842105263 19	52.631578947368396
MTLS	53.0612244897959	53.0612244897959	57.142857142857096	51.020408163265294	44.8979591836735	55.1020408163265	55.1020408163265	53.0612244897959	57.142857142857096
PS	45.789473684210 5	45.263157894736 8	44.736842105263 2	43.157894736842 1	49.473684210526 3	45.789473684210 5	46.842105263157904	45.789473684210 5	44.736842105263 2
PTT	58.9473684210526	58.9473684210526	52.631578947368396	58.9473684210526	58.9473684210526	55.789473684210 5	60.0	59.4736842105263	52.631578947368396
PTTEP	52.1052631578947	52.1052631578947	52.1052631578947	51.578947368421105	52.631578947368396	51.578947368421105	52.1052631578947	52.1052631578947	52.1052631578947
PTTGC	51.052631578947405	50.5263157894737	52.631578947368396	50.5263157894737	51.052631578947405	54.2105263157895	50.5263157894737	50.5263157894737	52.631578947368396
ROBINS	61.052631578947405	61.578947368421 1	46.315789473684 2	49.473684210526 3	56.842105263157904	45.263157894736 8	60.526315789473 69	61.578947368421 1	46.315789473684 2
SAWAD	38.88888888888889	38.88888888888889	41.11111111111111	38.88888888888889	37.77777777777778	38.88888888888889	38.88888888888889	41.11111111111111	
SCB	56.842105263157904	58.421052631578995	53.6842105263158	56.315789473684 2	57.894736842105296	52.631578947368396	56.315789473684 2	58.421052631578995	53.684210526315 8
SCC	53.6842105263158	55.2631578947368	57.368421052631604	52.631578947368396	55.2631578947368	53.157894736842 1	53.157894736842 1	54.736842105263 19	57.368421052631604
TASCO	45.263157894736 8	46.842105263157904	48.421052631578895	46.315789473684 2	45.789473684210 5	47.8947368421053	46.315789473684 2	46.315789473684 2	48.421052631578895
TCAP	49.473684210526 3	51.578947368421105	51.578947368421105	52.1052631578947	53.157894736842 1	52.631578947368396	50.0	51.578947368421105	51.578947368421105
TMB	52.631578947368396	54.2105263157895	50.0	54.2105263157895	51.052631578947405	51.052631578947405	52.1052631578947	54.2105263157895	49.4736842105263
TOP	60.526315789473 69	58.421052631578995	45.789473684210 5	59.4736842105263	57.894736842105296	46.842105263157904	60.526315789473 69	58.421052631578995	45.789473684210 5
TPIPL	50.0	51.052631578947405	52.631578947368396	47.8947368421053	46.842105263157904	50.5263157894737	49.4736842105263	50.5263157894737	52.631578947368396
TRUE	56.842105263157904	57.368421052631604	59.4736842105263	55.789473684210 5	53.684210526315 8	57.368421052631604	56.842105263157904	57.368421052631604	60.0
TTW	46.315789473684 2	43.157894736842 1	48.9473684210526	41.578947368421 1	41.052631578947 4	44.2105263157895	46.315789473684 2	43.157894736842 1	48.9473684210526
TU	59.473684210526 3	59.4736842105263	57.368421052631604	60.0	60.0	57.368421052631604	59.473684210526 3	59.473684210526 3	57.368421052631604
WHA	47.368421052631 6	46.842105263157904	52.631578947368396	47.8947368421053	47.8947368421053	47.8947368421053	47.8947368421053	47.368421052631 6	53.157894736842 1

Table 98 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.2105263157895	43.684210526315795	51.578947368421105	44.2105263157895	44.2105263157895	50.0	44.2105263157895	43.684210526315795	51.578947368421105
AOT	47.8947368421053	52.1052631578947	48.421052631578895	52.1052631578947	52.631578947368396	50.5263157894737	47.8947368421053	53.1578947368421	48.421052631578895
BA	46.296296296296305	38.88888888888889	50.0	42.592592592592595	42.592592592592595	42.592592592592595	46.296296296296305	38.88888888888889	50.0
BANPU	58.9473684210526	58.421052631578995	52.1052631578947	58.421052631578995	58.421052631578995	57.368421052631604	58.9473684210526	58.421052631578995	52.1052631578947
BBL	45.7894736842105	46.842105263157904	45.7894736842105	48.9473684210526	47.368421052631 6	47.8947368421053	45.263157894736 8	46.842105263157904	45.263157894736 8
BCP	48.9473684210526	50.5263157894737	55.2631578947368	48.421052631578895	49.473684210526 3	52.1052631578947	48.9473684210526	50.5263157894737	55.2631578947368
BDMS	55.263157894736 8	60.0	54.2105263157895	53.1578947368421	58.421052631578995	56.842105263157904	52.631578947368396	58.9473684210526	53.684210526315 8
BEC	57.368421052631604	55.2631578947368	56.842105263157904	57.894736842105296	55.7894736842105	57.894736842105296	57.368421052631604	55.263157894736 8	56.842105263157904
BH	46.315789473684 2	45.263157894736 8	47.8947368421053	46.842105263157904	46.315789473684 2	46.842105263157904	46.315789473684 2	45.263157894736 8	47.8947368421053
BLA	50.5263157894737	51.052631578947405	50.0	51.052631578947405	51.052631578947405	50.0	50.5263157894737	51.052631578947405	50.0
BTS	58.9473684210526	58.9473684210526	58.9473684210526	60.0	58.9473684210526	56.315789473684 2	58.9473684210526	59.4736842105263	58.9473684210526
CBG	56.000000000000 01	57.99999999999999	57.99999999999999	64.0	68.0	62.0	56.000000000000 01	62.0	54.0
CENTEL	56.842105263157904	58.9473684210526	54.736842105263 19	56.315789473684 2	58.421052631578995	56.842105263157904	57.368421052631604	58.9473684210526	54.736842105263 19
CK	54.2105263157895	53.1578947368421	53.1578947368421	54.2105263157895	54.2105263157895	52.1052631578947	55.2631578947368	54.2105263157895	53.1578947368421
CPALL	54.736842105263 19	52.631578947368396	60.0	51.578947368421105	48.421052631578895	51.578947368421105	54.736842105263 19	53.684210526315 8	58.9473684210526
CPF	53.1578947368421	50.5263157894737	51.052631578947405	49.473684210526 3	49.473684210526 3	47.368421052631 6	53.684210526315 8	50.0	51.052631578947405
CPN	51.578947368421105	53.1578947368421	56.842105263157904	55.7894736842105	54.736842105263 19	51.578947368421105	50.0	53.684210526315 8	56.842105263157904
DELTA	54.2105263157895	54.2105263157895	54.736842105263 19	55.7894736842105	53.684210526315 8	54.2105263157895	54.2105263157895	54.2105263157895	54.736842105263 19
DTAC	47.8947368421053	48.421052631578895	53.157894736842 1	48.9473684210526	48.421052631578895	48.9473684210526	47.8947368421053	48.9473684210526	53.157894736842 1
EGCO	48.421052631578895	48.9473684210526	53.157894736842 1	51.052631578947405	48.421052631578895	54.736842105263 19	50.0	48.9473684210526	54.736842105263 19
GLOW	48.421052631578895	46.842105263157904	48.421052631578895	52.1052631578947	47.8947368421053	50.5263157894737	49.473684210526 3	46.842105263157904	48.421052631578895
GPSC	43.75	43.75	43.75	43.75	37.5	43.75	43.75	43.75	43.75
HMPRO	56.315789473684 2	59.4736842105263	46.842105263157904	58.9473684210526	58.9473684210526	50.0	56.315789473684 2	59.4736842105263	47.368421052631 6
INTUCH	45.263157894736 8	45.263157894736 8	52.1052631578947	44.7368421052632	47.8947368421053	50.5263157894737	45.263157894736 8	45.263157894736 8	52.1052631578947
IRPC	53.6842105263158	54.736842105263 19	55.7894736842105	53.157894736842 1	53.157894736842 1	52.631578947368396	54.2105263157895	55.263157894736 8	55.7894736842105
IVL	54.2105263157895	50.0	48.421052631578895	53.157894736842 1	48.9473684210526	52.1052631578947	54.2105263157895	50.0	48.421052631578895
KBANK	44.2105263157895	44.736842105263 2	45.263157894736 8	44.736842105263 2	43.157894736842 1	46.842105263157904	44.736842105263 2	44.736842105263 2	45.263157894736 8
KCE	47.8947368421053	45.7894736842105	54.736842105263 19	50.5263157894737	46.315789473684 2	50.5263157894737	48.9473684210526	45.263157894736 8	54.736842105263 19
KTB	47.368421052631 6	46.842105263157904	50.0	47.368421052631 6	46.842105263157904	48.9473684210526	47.368421052631 6	46.315789473684 2	50.0
LH	60.0	60.526315789473 69	61.57894736842 11	60.0	59.4736842105263	60.0	61.052631578947405	60.526315789473 69	61.57894736842 11
MINT	51.578947368421105	54.736842105263 19	55.263157894736 8	48.9473684210526	52.1052631578947	48.9473684210526	52.1052631578947	53.684210526315 8	53.157894736842 1
MTLS	53.0612244897959	53.0612244897959	57.142857142857096	51.020408163265294	44.8979591836735	55.1020408163265	55.1020408163265	53.0612244897959	57.142857142857096
PS	44.2105263157895	44.2105263157895	43.157894736842 1	42.105263157894704	46.315789473684 2	42.105263157894704	43.684210526315795	45.263157894736 8	43.157894736842 1
PTT	59.4736842105263	59.4736842105263	52.631578947368396	58.9473684210526	58.9473684210526	55.789473684210 5	59.4736842105263	58.9473684210526	52.631578947368396
PTTEP	52.631578947368396	52.631578947368396	53.157894736842 1	52.1052631578947	52.631578947368396	52.631578947368396	52.631578947368396	52.631578947368396	53.157894736842 1
PTTGC	51.052631578947405	50.5263157894737	52.631578947368396	51.052631578947405	50.5263157894737	54.2105263157895	50.5263157894737	50.5263157894737	52.631578947368396
ROBINS	60.0	61.578947368421 1	46.315789473684 2	50.5263157894737	57.368421052631604	45.263157894736 8	60.526315789473 69	61.578947368421 1	46.315789473684 2
SAWAD	38.88888888888889	40.0	41.11111111111111	38.88888888888889	37.77777777777778	38.88888888888889	38.88888888888889	41.11111111111111	
SCB	56.842105263157904	57.894736842105296	53.6842105263158	55.263157894736 8	57.894736842105296	52.631578947368396	56.842105263157904	58.421052631578995	53.684210526315 8
SCC	54.2105263157895	55.789473684210 5	56.842105263157904	51.578947368421105	54.736842105263 19	53.157894736842 1	53.157894736842 1	55.263157894736 8	56.842105263157904
TASCO	42.105263157894704	39.473684210526 3	44.2105263157895	42.105263157894704	40.52631578947 37	42.631578947368396	41.578947368421 1	39.473684210526 3	43.684210526315795
TCAP	49.473684210526 3	51.578947368421105	51.578947368421105	52.1052631578947	52.631578947368396	52.631578947368396	49.473684210526 3	51.578947368421105	51.578947368421105
TMB	51.578947368421105	54.2105263157895	51.578947368421105	53.684210526315 8	50.5263157894737	51.052631578947405	51.578947368421105	54.2105263157895	51.052631578947405
TOP	60.526315789473 69	58.421052631578995	45.789473684210 5	59.473684210526 3	57.894736842105296	46.842105263157904	58.9473684210526	58.421052631578995	45.789473684210 5
TPIPL	50.0	49.4736842105263	50.5263157894737	50.0	49.4736842105263	51.052631578947405	50.5263157894737	50.5263157894737	50.5263157894737
TRUE	56.315789473684 2	58.421052631578995	56.315789473684 2	56.842105263157904	58.421052631578995	56.842105263157904	55.263157894736 8	58.421052631578995	56.315789473684 2
TTW	45.789473684210 5	44.736842105263 2	48.421052631578895	41.578947368421 1	42.631578947368396	44.736842105263 2	45.789473684210 5	44.2105263157895	48.421052631578895
TU	59.473684210526 3	59.473684210526 3	57.368421052631604	60.0	60.0	57.368421052631604	59.473684210526 3	59.473684210526 3	57.368421052631604
WHA	46.842105263157904	47.8947368421053	52.1052631578947	48.421052631578895	47.8947368421053	46.842105263157904	46.842105263157904	47.368421052631 6	52.1052631578947

Table 99 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	48.9473684210526	41.0526315789474	44.2105263157895	54.2105263157895	43.684210526315795	43.684210526315795	48.9473684210526
AOT	48.9473684210526	54.736842105263 19	53.1578947368421	50.0	53.68421052631 58	52.10526315789 47	51.052631578947405	55.78947368421 05	53.1578947368421
BA	38.88888888888889	38.88888888888889	37.037037037037	42.592592592592 95	38.88888888888889	46.296296296296 305	38.88888888888889	38.88888888888889	37.037037037037
BANPU	58.9473684210526	58.9473684210526	54.736842105263 19	50.5263157894737	58.421052631578995	49.4736842105263	58.9473684210526	59.4736842105263	54.736842105263 19
BBL	51.052631578947405	50.5263157894737	53.68421052631 58	55.2631578947368	52.631578947368396	47.3684210526316	51.052631578947405	51.052631578947405	53.68421052631 58
BCP	44.7368421052632	43.1578947368421	45.7894736842105	51.052631578947405	47.3684210526316	46.3157894736842	45.2631578947368	43.684210526315795	45.7894736842105
BDMS	49.4736842105263	53.1578947368421	50.5263157894737	48.9473684210526	53.1578947368421	54.736842105263 19	50.0	53.68421052631 58	50.5263157894737
BEC	54.736842105263 19	54.736842105263 19	56.3157894736842	57.894736842105 296	55.2631578947368	54.2105263157895	54.736842105263 19	54.2105263157895	56.3157894736842
BH	47.8947368421053	48.9473684210526	47.8947368421053	50.0	47.3684210526316	45.2631578947368	47.8947368421053	48.9473684210526	47.8947368421053
BLA	50.0	51.052631578947405	48.421052631578895	46.3157894736842	51.052631578947405	52.631578947368396	50.5263157894737	50.5263157894737	48.421052631578895
BTS	54.2105263157895	53.68421052631 58	55.2631578947368	61.052631578947405	54.736842105263 19	56.842105263157904	54.736842105263 19	53.68421052631 58	54.736842105263 19
CBG	62.0	62.0	57.999999999999 99	57.999999999999 99	62.0	54.0	62.0	62.0	62.0
CENTEL	53.68421052631 58	53.1578947368421	54.736842105263 19	56.3157894736842	56.3157894736842	48.9473684210526	53.68421052631 58	53.68421052631 58	54.736842105263 19
CK	47.8947368421053	46.842105263157904	44.2105263157895	50.0	50.5263157894737	49.4736842105263	47.3684210526316	46.3157894736842	44.2105263157895
CPALL	46.842105263157904	49.4736842105263	52.1052631578947	52.1052631578947	47.3684210526316	50.0	48.9473684210526	50.0	52.1052631578947
CPF	51.578947368421105	52.1052631578947	48.9473684210526	50.0	51.578947368421105	48.9473684210526	50.5263157894737	51.578947368421105	48.9473684210526
CPN	50.0	51.052631578947405	48.421052631578895	44.7368421052632	43.1578947368421	50.5263157894737	50.0	51.052631578947405	48.9473684210526
DELTA	50.0	50.0	50.5263157894737	53.6842105263158	53.1578947368421	51.578947368421105	50.0	50.0	50.5263157894737
DTAC	51.052631578947405	47.8947368421053	50.0	45.7894736842105	46.842105263157904	50.5263157894737	51.578947368421105	47.8947368421053	50.0
EGCO	40.0	40.5263157894737	46.842105263157904	53.1578947368421	45.7894736842105	47.8947368421053	40.0	40.5263157894737	46.842105263157904
GLOW	44.2105263157895	42.631578947368396	45.2631578947368	39.4736842105263	42.105263157894704	47.3684210526316	44.2105263157895	42.631578947368396	45.2631578947368
GPSC	43.75	43.75	43.75	50.0	50.0	56.25	43.75	50.0	43.75
HMPRO	52.631578947368396	50.0	55.7894736842105	49.4736842105263	53.6842105263158	46.3157894736842	52.1052631578947	50.0	55.2631578947368
INTUCH	42.105263157894704	41.578947368421 1	45.2631578947368	45.2631578947368	43.1578947368421	51.578947368421105	43.1578947368421	40.5263157894737	45.2631578947368
IRPC	51.052631578947405	50.0	52.1052631578947	50.0	49.4736842105263	53.1578947368421	51.578947368421105	50.0	52.1052631578947
IVL	42.631578947368396	40.0	42.631578947368396	51.578947368421105	40.5263157894737	54.2105263157895	41.052631578974	40.5263157894737	42.631578947368396
KBANK	46.3157894736842	46.3157894736842	44.7368421052632	48.421052631578895	45.2631578947368	48.421052631578895	46.3157894736842	46.3157894736842	44.7368421052632
KCE	45.2631578947368	42.631578947368396	44.7368421052632	45.7894736842105	43.1578947368421	43.684210526315795	44.7368421052632	42.631578947368396	44.7368421052632
KTB	42.631578947368396	38.9473684210526	41.5789473684211	57.894736842105296	47.8947368421053	41.5789473684211	41.052631578974	38.9473684210526	41.5789473684211
LH	59.4736842105263	53.684210526315 8	59.4736842105263	55.2631578947368	54.736842105263 19	44.2105263157895	58.9473684210526	53.684210526315 8	59.4736842105263
MINT	48.9473684210526	49.4736842105263	49.4736842105263	46.842105263157904	46.3157894736842	46.842105263157904	50.5263157894737	50.0	49.4736842105263
MTLS	48.9795918367347	46.9387755102041	46.9387755102041	51.020408163265294	48.9795918367347	61.2244897959184	44.8979591836735	46.9387755102041	46.9387755102041
PS	38.9473684210526	39.4736842105263	41.052631578974	44.7368421052632	40.5263157894737	44.7368421052632	38.9473684210526	40.0	41.052631578974
PTT	55.7894736842105	56.842105263157904	61.052631578947405	51.5789473684211 05	56.3157894736842	44.2105263157895	56.842105263157904	56.3157894736842	61.052631578947405
PTTEP	52.1052631578947	52.1052631578947	51.578947368421105	47.8947368421053	52.1052631578947	51.052631578947405	52.1052631578947	52.1052631578947	51.578947368421105
PTTGC	46.3157894736842	46.842105263157904	51.052631578947405	46.3157894736842	50.0	52.1052631578947	46.3157894736842	46.842105263157904	51.052631578947405
ROBINS	50.0	52.1052631578947	43.1578947368421	56.842105263157904	55.263157894736 8	43.1578947368421	52.1052631578947	54.2105263157895	43.1578947368421
SAWAD	40.0	41.111111111111 1	42.2222222222222	37.7777777777778	40.0	37.7777777777778	40.0	41.111111111111 1	41.111111111111 1
SCB	51.578947368421105	53.1578947368421	50.0	51.052631578947405	53.1578947368421	47.8947368421053	52.1052631578947	54.2105263157895	50.0
SCC	56.3157894736842	55.7894736842105	57.368421052631604	52.631578947368396	52.1052631578947	54.736842105263 19	54.736842105263 19	55.7894736842105	57.368421052631604
TASCO	42.105263157894704	40.5263157894737	43.1578947368421	40.5263157894737	40.0	42.105263157894704	41.5789473684211	40.5263157894737	43.1578947368421
TCAP	46.3157894736842	47.3684210526316	44.7368421052632	45.2631578947368	46.842105263157904	52.631578947368396	46.3157894736842	47.3684210526316	44.2105263157895
TMB	40.0	43.1578947368421	41.5789473684211	50.5263157894737	48.9473684210526	42.105263157894704	43.1578947368421	41.5789473684211	41.5789473684211
TOP	57.894736842105296	56.842105263157904	44.7368421052632	55.2631578947368	58.421052631578995	46.3157894736842	58.421052631578995	56.842105263157904	44.7368421052632
TPIPL	44.2105263157895	42.105263157894704	46.3157894736842	47.8947368421053	47.3684210526316	50.0	43.1578947368421	43.1578947368421	45.2631578947368
TRUE	60.0	56.842105263157904	57.894736842105296	55.7894736842105	56.842105263157904	54.2105263157895	57.894736842105296	56.842105263157904	57.894736842105296
TTW	37.894736842105296	34.736842105232	44.736842105232	42.105263157894704	42.631578947368396	44.736842105232	39.4736842105263	35.263157894736 8	44.2105263157895
TU	57.894736842105296	56.842105263157904	57.894736842105296	55.2631578947368	57.894736842105296	53.6842105263158	57.894736842105296	56.3157894736842	57.894736842105296
WHA	46.3157894736842	46.3157894736842	47.3684210526316	45.2631578947368	45.2631578947368	44.2105263157895	46.842105263157904	46.3157894736842	47.3684210526316

PredictDirection13 : EMD of closing change

Table 100 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.2105263157895	43.684210526315795	44.736842105232	43.684210526315795	43.684210526315795	44.2105263157895	44.2105263157895	43.684210526315795	44.736842105232
AOT	56.842105263157904	61.052631578947405	56.3157894736842	56.842105263157904	60.0	55.7894736842105	56.842105263157904	60.526315789473 69	56.3157894736842
BA	61.111111111111 1	61.111111111111 1	61.111111111111 1	59.2592592592593	50.0	59.2592592592593	59.2592592592593	59.2592592592593	61.111111111111 1
BANPU	42.631578947368396	42.105263157894704	41.5789473684211	41.5789473684211	41.5789473684211	41.5789473684211	42.105263157894704	41.5789473684211	41.5789473684211
BBL	44.2105263157895	45.2631578947368	43.684210526315795	45.7894736842105	44.736842105232	45.2631578947368	44.2105263157895	45.2631578947368	43.684210526315795
BCP	43.1578947368421	45.2631578947368	45.7894736842105	44.2105263157895	41.5789473684211	48.9473684210526	43.1578947368421	44.2105263157895	45.7894736842105
BDMS	61.052631578947405	59.4736842105263	61.052631578947405	61.052631578947405	60.0	61.052631578947405	61.052631578947405	60.0	54.2105263157895
BEC	45.2631578947368	45.2631578947368	45.2631578947368	45.2631578947368	45.2631578947368	45.2631578947368	45.2631578947368	45.2631578947368	45.2631578947368
BH	59.4736842105263	52.631578947368396	57.894736842105296	58.9473684210526	53.684210526315 8	56.842105263157904	58.9473684210526	53.684210526315 8	56.842105263157904
BLA	48.421052631578895	46.842105263157904	47.8947368421053	47.3684210526316	48.421052631578895	48.9473684210526	48.421052631578895	46.842105263157904	47.8947368421053
BTS	47.3684210526316	50.0	42.631578947368396	48.421052631578895	49.4736842105263	44.736842105232	47.8947368421053	48.421052631578895	43.1578947368421
CBG	70.0	70.0	70.0	70.0	68.0	70.0	70.0	70.0	70.0
CENTEL	57.368421052631604	59.4736842105263	55.2631578947368	57.894736842105296	60.526315789473 69	55.7894736842105	57.368421052631604	60.0	55.2631578947368
CK	43.684210526315795	48.9473684210526	56.842105263157904	43.1578947368421	48.421052631578895	57.894736842105296	43.684210526315795	49.4736842105263	56.3157894736842
CPALL	52.631578947368396	56.3157894736842	48.421052631578895	53.1578947368421	53.684210526315 8	50.5263157894737	52.631578947368396	56.3157894736842	48.421052631578895
CPF	46.3157894736842	45.7894736842105	49.4736842105263	48.421052631578895	45.2631578947368	48.421052631578895	46.842105263157904	45.2631578947368	49.4736842105263
CPN	53.684210526315 8	54.736842105263 19	55.2631578947368	54.736842105263 19	55.2631578947368	52.631578947368396	53.1578947368421	54.736842105263 19	55.7894736842105
DELTA	56.842105263157904	58.9473684210526	53.6842105263158	57.368421052631604	59.4736842105263	55.2631578947368	57.368421052631604	59.4736842105263	54.2105263157895
DTAC	50.0	50.0	48.9473684210526	50.0	50.0	49.4736842105263	50.0	50.0	48.9473684210526
EGCO	56.842105263157904	57.894736842105296	56.3157894736842	56.842105263157904	57.894736842105296	55.7894736842105	57.368421052631604	58.421052631578995	56.3157894736842
GLOW	48.9473684210526	47.3684210526316	48.9473684210526	47.8947368421053	48.9473684210526	47.8947368421053	48.9473684210526	47.3684210526316	48.9473684210526
GPSC	50.0	50.0	43.75	43.75	56.25	43.75	43.75	56.25	43.75
HMPRO	53.1578947368421	52.1052631578947	51.578947368421105	52.631578947368396	53.1578947368421	51.052631578947405	53.1578947368421	53.1578947368421	51.578947368421105
INTUCH	43.1578947368421	42.105263157894704	42.105263157894704	42.105263157894704	42.631578947368396	42.105263157894704	43.1578947368421	42.105263157894704	42.105263157894704
IRPC	58.9473684210526	58.421052631578995	60.0	57.894736842105296	57.368421052631604	58.9473684210526	58.421052631578995	58.9473684210526	60.0
IVL	51.052631578947405	51.578947368421105	57.894736842105296	52.1052631578947	53.1578947368421	51.578947368421105	52.1052631578947	52.1052631578947	57.894736842105296
KBANK	46.3157894736842	46.3157894736842	47.3684210526316	46.3157894736842	46.3157894736842	45.2631578947368	45.7894736842105	46.842105263157904	47.3684210526316
KCE	59.4736842105263	60.526315789473 69	60.0	60.526315789473 69	60.526315789473 69	60.0	59.4736842105263	60.526315789473 69	60.0
KTB	50.5263157894737	48.421052631578895	47.8947368421053	48.9473684210526	48.421052631578895	48.421052631578895	51.052631578947405	47.8947368421053	47.8947368421053
LH	50.0	47.3684210526316	51.578947368421105	53.1578947368421	47.8947368421053	54.2105263157895	50.0	47.3684210526316	52.1052631578947
MINT	59.4736842105263	60.0	56.842105263157904	59.4736842105263	60.0	58.9473684210526	59.4736842105263	60.0	56.842105263157904
MTLS	53.0612244897959	51.020408163265294	55.1020408163265	48.9795918367347	48.9795918367347	48.9795918367347	51.020408163265294	55.1020408163265	
PS	52.631578947368396	52.631578947368396	56.3157894736842	50.0	53.1578947368421	52.631578947368396	52.631578947368396	52.631578947368396	56.3157894736842
PTT	45.2631578947368	44.736842105232	45.7894736842105	46.3157894736842	44.2105263157895	45.2631578947368	46.3157894736842	44.736842105232	46.3157894736842
PTTEP	47.8947368421053	47.3684210526316	45.2631578947368	47.8947368421053	46.842105263157904	46.842105263157904	48.421052631578895	46.842105263157904	45.2631578947368
PTTGC	49.4736842105263	50.0	49.4736842105263	50.5263157894737	51.052631578947405	53.6842105263158	50.0	50.5263157894737	48.421052631578895
ROBINS	54.2105263157895	50.0	53.6842105263158	54.736842105263 19	49.4736842105263	56.3157894736842	54.2105263157895	50.0	53.6842105263158
SAWAD	50.0	46.6666666666667	46.6666666666667	48.8888888888889	42.2222222222222	50.0	50.0	46.6666666666667	46.6666666666667
SCB	40.5263157894737	40.5263157894737	42.631578947368396	40.5263157894737	40.5263157894737	40.5263157894737	40.5263157894737	40.5263157894737	43.1578947368421
SCC	48.421052631578895	51.052631578947405	46.3157894736842	48.9473684210526	52.1052631578947	47.3684210526316	48.421052631578895	51.052631578947405	46.842105263157904
TASCO	54.2105263157895	56.3157894736842	44.2105263157895	51.052631578947405	56.842105263157904	53.6842105263158	54.736842105263 19	56.842105263157904	43.1578947368421
TCAP	52.1052631578947	53.1578947368421	53.6842105263158	53.1578947368421	54.2105263157895	51.578947368421105	52.631578947368396	53.6842105263158	53.6842105263158
TMB	47.3684210526316	48.421052631578895	57.368421052631604	46.842105263157904	48.421052631578895	50.0	47.3684210526316	48.421052631578895	57.368421052631604
TOP	58.9473684210526	59.4736842105263	60.0	57.894736842105296	60.0	60.526315789473 69	58.9473684210526	59.4736842105263	60.0
TPIPL	56.3157894736842	58.421052631578995	54.2105263157895	57.894736842105296	57.894736842105296	54.2105263157895	56.842105263157904	58.421052631578995	53.1578947368421
TRUE	48.9473684210526	48.421052631578895	48.9473684210526	48.9473684210526	48.9473684210526	47.3684210526316	49.4736842105263	48.421052631578895	51.578947368421105
TTW	42.105263157894704	43.684210526315795	38.9473684210526	42.105263157894704	41.5789473684211	46.3157894736842	41.5789473684211	43.1578947368421	38.9473684210526
TU	55.2631578947368	54.736842105263 19	53.6842105263158	56.3157894736842	52.1052631578947	54.2105263157895	55.7894736842105	54.736842105263 19	53.6842105263158
WHA	44.2105263157895	43.684210526315795	44.736842105232	44.736842105232	43.684210526315795	43.684210526315795	44.2105263157895	43.684210526315795	45.7894736842105

Table 101 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.2105263157895	45.7894736842105	50.5263157894737	49.4736842105263	45.2631578947368	52.6315789473683396	41.0526315789474	46.3157894736842	50.0
AOT	53.6842105263158	54.2105263157895	55.2631578947368	53.6842105263158	51.5789473684211105	52.1052631578947	54.2105263157895	54.2105263157895	55.7894736842105
BA	59.2592592592593	61.1111111111111	59.2592592592593	57.4074074074074	59.2592592592593	59.2592592592593	59.2592592592593	59.2592592592593	59.2592592592593
BANPU	47.3684210526316	55.7894736842105	48.9473684210526	49.4736842105263	55.7894736842105	45.2631578947368	48.4210526315789895	56.3157894736842	48.9473684210526
BBL	48.4210526315789895	47.8947368421053	51.5789473684211105	48.9473684210526	49.4736842105263	48.4210526315789895	48.4210526315789895	47.8947368421053	51.5789473684211105
BCP	45.2631578947368	45.7894736842105	47.8947368421053	48.4210526315789895	47.3684210526316	51.0526315789474105	46.8421052631579904	46.8421052631579904	47.8947368421053
BDMS	35.2631578947368	36.8421052631579	43.6842105263157795	38.9473684210526	35.7894736842105	48.9473684210526	36.3157894736842	36.8421052631579	43.6842105263157795
BEC	44.2105263157895	43.6842105263157795	51.5789473684211105	43.6842105263157795	44.736842105263632	48.4210526315789895	44.2105263157895	43.6842105263157795	51.5789473684211105
BH	54.7368421052619	52.1052631578947	54.2105263157895	54.2105263157895	56.3157894736842	53.1578947368421	53.6842105263158	52.1052631578947	54.2105263157895
BLA	41.5789473684211	42.6315789473683396	46.3157894736842	43.1578947368421	43.1578947368421	42.6315789473683396	41.5789473684211	43.1578947368421	46.3157894736842
BTS	38.9473684210526	36.3157894736842	44.2105263157895	41.0526315789474	38.9473684210526	45.2631578947368	38.9473684210526	36.3157894736842	44.2105263157895
CBG	40.0	46.0	40.0	50.0	46.0	50.0	42.0	46.0	40.0
CENTEL	54.2105263157895	55.2631578947368	54.2105263157895	53.1578947368421	52.6315789473683396	51.5789473684211105	54.2105263157895	54.7368421052619	54.2105263157895
CK	42.1052631578947104	42.6315789473683396	44.2105263157895	39.4736842105263	40.0	38.9473684210526	42.1052631578947104	43.1578947368421	44.7368421052632
CPALL	45.2631578947368	47.8947368421053	44.7368421052632	46.8421052631579904	46.8421052631579904	41.5789473684211	45.2631578947368	48.4210526315789895	44.2105263157895
CPF	54.2105263157895	53.1578947368421	50.5263157894737	50.0	54.7368421052619	48.9473684210526	54.2105263157895	53.1578947368421	50.0
CPN	41.5789473684211	41.0526315789474	41.0526315789474	42.6315789473683396	42.6315789473683396	44.7368421052632	41.5789473684211	41.0526315789474	42.1052631578947104
DELTA	53.1578947368421	53.1578947368421	52.1052631578947	51.0526315789474105	52.1052631578947	50.5263157894737	53.1578947368421	53.1578947368421	52.6315789473683396
DTAC	48.4210526315789895	47.3684210526316	49.4736842105263	51.5789473684211105	50.5263157894737	55.7894736842105	48.4210526315789895	47.3684210526316	49.4736842105263
EGCO	46.8421052631579904	53.6842105263158	52.6315789473683396	53.6842105263158	57.3684210526316004	45.2631578947368	47.8947368421053	54.2105263157895	52.6315789473683396
GLOW	45.7894736842105	47.3684210526316	51.5789473684211105	49.4736842105263	50.5263157894737	54.2105263157895	45.7894736842105	46.8421052631579904	51.5789473684211105
GPSC	56.25	56.25	62.5	50.0	43.75	50.0	56.25	56.25	62.5
HMPRO	57.3684210526316004	56.3157894736842	48.4210526315789895	41.0526315789474	50.0	54.2105263157895	56.3157894736842	55.7894736842105	48.4210526315789895
INTUCH	51.0526315789474105	52.6315789473683396	53.6842105263158	51.5789473684211105	51.5789473684211105	52.6315789473683396	51.0526315789474105	53.6842105263158	54.2105263157895
IRPC	50.5263157894737	54.2105263157895	45.7894736842105	55.2631578947368	55.2631578947368	48.9473684210526	50.5263157894737	54.2105263157895	46.3157894736842
IVL	51.5789473684211105	50.0	62.1052631578947	52.1052631578947	50.0	61.0526315789474105	52.6315789473683396	50.0	62.6315789473684
KBANK	51.5789473684211105	53.6842105263158	52.6315789473683396	49.4736842105263	49.4736842105263	50.5263157894737	50.0	53.6842105263158	52.6315789473683396
KCE	58.9473684210526	57.3684210526316004	51.5789473684211105	59.4736842105263	56.3157894736842	51.0526315789474105	56.84210526315704	57.8947368421052096	51.5789473684211105
KTB	41.5789473684211	43.6842105263157795	50.5263157894737	43.1578947368421	42.1052631578947104	46.3157894736842	41.5789473684211	43.6842105263157795	50.0
LH	46.3157894736842	44.7368421052632	43.6842105263157795	45.2631578947368	45.7894736842105	45.2631578947368	46.3157894736842	45.2631578947368	43.6842105263157795
MINT	44.2105263157895	43.1578947368421	50.0	52.6315789473683396	49.4736842105263	52.1052631578947	44.7368421052632	43.6842105263157795	50.0
MTLS	63.2653061224490005	65.3061224489796	63.2653061224490005	55.1020408163265	57.1428571428575096	63.2653061224490005	61.2244897959184	65.3061224489796	63.2653061224490005
PS	40.5263157894737	36.8421052631579	46.8421052631579904	40.0	35.2631578947368	42.1052631578947104	39.4736842105263	36.3157894736842	46.8421052631579904
PTT	57.8947368421052096	55.7894736842105	47.8947368421053	56.3157894736842	55.2631578947368	52.1052631578947	57.8947368421052096	55.7894736842105	47.3684210526316
PTTEP	52.6315789473683396	52.1052631578947	48.4210526315789895	52.6315789473683396	52.6315789473683396	50.5263157894737	52.6315789473683396	51.5789473684211105	48.4210526315789895
PTTGC	48.4210526315789895	48.9473684210526	53.6842105263158	52.6315789473683396	50.5263157894737	57.8947368421052096	47.8947368421053	50.5263157894737	53.6842105263158
ROBINS	48.4210526315789895	48.4210526315789895	44.2105263157895	41.5789473684211	46.3157894736842	40.5263157894737	48.4210526315789895	48.4210526315789895	44.2105263157895
SAWAD	55.5555555555556	53.3333333333333	50.0	50.0	55.5555555555556	52.2222222222222	52.2222222222222	53.3333333333333	48.8888888888889
SCB	60.5263157894737369	57.3684210526316004	41.0526315789474	61.0526315789474105	59.4736842105263	40.0	60.5263157894737369	56.8421052631579004	41.0526315789474
SCC	44.7368421052632	47.8947368421053	50.0	54.7368421052619	53.1578947368421	52.6315789473683396	46.8421052631579904	46.8421052631579904	50.0
TASCO	51.0526315789474105	50.5263157894737	48.4210526315789895	51.5789473684211105	51.5789473684211105	51.0526315789474105	51.5789473684211105	48.9473684210526	47.8947368421053
TCAP	45.2631578947368	45.7894736842105	50.0	45.2631578947368	46.3157894736842	47.3684210526316	45.2631578947368	46.3157894736842	50.0
TMB	50.0	55.7894736842105	55.7894736842105	52.6315789473683396	55.7894736842105	47.3684210526316	48.9473684210526	55.2631578947368	54.7368421052619
TOP	42.6315789473683396	42.1052631578947104	52.1052631578947	41.5789473684211	45.2631578947368	44.7368421052632	42.6315789473683396	42.1052631578947104	51.0526315789474105
TPIPL	43.6842105263157795	41.0526315789474	55.7894736842105	50.5263157894737	48.4210526315789895	50.0	43.1578947368421	40.5263157894737	53.1578947368421
TRUE	42.1052631578947104	45.7894736842105	48.9473684210526	49.4736842105263	46.3157894736842	47.8947368421053	42.1052631578947104	46.3157894736842	48.9473684210526
TTW	53.1578947368421	46.3157894736842	47.8947368421053	47.3684210526316	40.0	51.5789473684211105	51.5789473684211105	45.7894736842105	47.8947368421053
TU	55.2631578947368	56.8421052631579004	52.6315789473683396	55.7894736842105	55.2631578947368	52.1052631578947	55.7894736842105	56.3157894736842	52.6315789473683396
WHA	47.8947368421053	44.2105263157895	50.5263157894737	45.2631578947368	45.2631578947368	50.0	52.6315789473683396	45.2631578947368	50.5263157894737

Table 102 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.2105263157895	45.7894736842105	53.6842105263158	47.8947368421053	48.4210526315789895	53.6842105263158	44.2105263157895	45.2631578947368	54.2105263157895
AOT	53.6842105263158	53.6842105263158	56.8421052631579004	52.1052631578947	52.1052631578947	51.0526315789474105	53.6842105263158	53.6842105263158	56.8421052631579004
BA	59.2592592592593	59.2592592592593	57.4074074074074	57.4074074074074	55.5555555555556	50.0	57.4074074074074	55.5555555555556	57.4074074074074
BANPU	47.3684210526316	56.3157894736842	48.4210526315789895	48.9473684210526	54.7368421052619	47.3684210526316	48.4210526315789895	55.7894736842105	51.0526315789474105
BBL	47.3684210526316	47.8947368421053	50.0	50.0	48.9473684210526	52.1052631578947	48.9473684210526	48.4210526315789895	49.4736842105263
BCP	45.2631578947368	45.7894736842105	47.8947368421053	47.3684210526316	47.3684210526316	51.0526315789474105	45.7894736842105	46.3157894736842	47.8947368421053
BDMS	36.3157894736842	36.8421052631579	43.6842105263157795	38.9473684210526	35.7894736842105	48.9473684210526	36.3157894736842	36.8421052631579	43.6842105263157795
BEC	44.2105263157895	43.6842105263157795	51.0526315789474105	44.2105263157895	44.736842105263632	49.4736842105263	44.2105263157895	43.6842105263157795	51.0526315789474105
BH	53.6842105263158	54.2105263157895	56.8421052631579004	51.5789473684211105	55.2631578947368	50.5263157894737	53.1578947368421	54.2105263157895	56.8421052631579004
BLA	41.5789473684211	42.6315789473683396	46.3157894736842	42.1052631578947104	43.1578947368421	42.6315789473683396	41.0526315789474	42.6315789473683396	46.3157894736842
BTS	37.8947368421052096	36.3157894736842	46.3157894736842	40.0	38.9473684210526	43.1578947368421	37.8947368421052096	36.3157894736842	46.3157894736842
CBG	42.0	44.0	40.0	50.0	46.0	50.0	42.0	42.0	40.0
CENTEL	54.2105263157895	55.2631578947368	52.1052631578947	53.6842105263158	51.5789473684211105	50.0	54.2105263157895	55.2631578947368	52.1052631578947
CK	40.5263157894737	40.5263157894737	45.2631578947368	39.4736842105263	40.0	40.0	40.5263157894737	40.5263157894737	45.2631578947368
CPALL	45.2631578947368	47.8947368421053	44.7368421052632	46.8421052631579904	46.8421052631579904	41.5789473684211	45.2631578947368	48.4210526315789895	44.2105263157895
CPF	55.2631578947368	53.6842105263158	54.2105263157895	48.9473684210526	50.0	53.1578947368421	55.2631578947368	53.6842105263158	54.2105263157895
CPN	41.5789473684211	40.5263157894737	41.0526315789474	42.6315789473683396	43.1578947368421	44.7368421052632	41.0526315789474	41.0526315789474	42.1052631578947104
DELTA	53.1578947368421	53.1578947368421	52.1052631578947	50.5263157894737	52.1052631578947	50.5263157894737	53.1578947368421	53.1578947368421	52.6315789473683396
DTAC	49.4736842105263	47.8947368421053	53.6842105263158	50.5263157894737	50.5263157894737	48.9473684210526	48.9473684210526	47.8947368421053	53.6842105263158
EGCO	47.8947368421053	53.6842105263158	52.6315789473683396	53.6842105263158	57.3684210526316004	45.2631578947368	47.3684210526316	54.2105263157895	52.6315789473683396
GLOW	44.7368421052632	47.8947368421053	52.1052631578947	51.0526315789474105	52.1052631578947	53.1578947368421	44.7368421052632	47.3684210526316	51.5789473684211105
GPSC	62.5	56.25	62.5	56.25	43.75	50.0	56.25	56.25	62.5
HMPRO	57.3684210526316004	56.3157894736842	48.9473684210526	40.0	50.5263157894737	53.6842105263158	56.3157894736842	55.2631578947368	48.9473684210526
INTUCH	51.5789473684211105	53.6842105263158	53.6842105263158	51.5789473684211105	51.5789473684211105	52.6315789473683396	50.5263157894737	53.1578947368421	54.2105263157895
IRPC	50.5263157894737	53.6842105263158	46.8421052631579904	54.2105263157895	55.7894736842105	48.4210526315789895	50.5263157894737	54.2105263157895	46.8421052631579904
IVL	50.0	49.4736842105263	58.9473684210526	46.3157894736842	50.0	57.8947368421052096	50.5263157894737	50.0	58.9473684210526
KBANK	50.5263157894737	51.0526315789474105	51.0526315789474105	48.9473684210526	48.4210526315789895	48.9473684210526	51.0526315789474105	51.0526315789474105	51.0526315789474105
KCE	57.3684210526316004	57.8947368421052096	53.1578947368421	55.2631578947368	56.3157894736842	53.1578947368421	56.84210526315704	58.4210526315789895	52.6315789473683396
KTB	41.5789473684211	43.1578947368421	50.5263157894737	44.2105263157895	42.1052631578947104	46.3157894736842	41.5789473684211	43.1578947368421	50.0
LH	46.3157894736842	45.2631578947368	44.7368421052632	44.2105263157895	44.2105263157895	44.2105263157895	46.3157894736842	45.7894736842105	44.7368421052632
MINT	43.6842105263157795	44.2105263157895	50.0	46.3157894736842	43.6842105263157795	51.0526315789474105	42.6315789473683396	43.6842105263157795	50.5263157894737
MTLS	59.1836734693878	65.3061224489796	63.2653061224490005	55.1020408163265	57.1428571428575096	63.2653061224490005	61.2244897959184	65.3061224489796	63.2653061224490005
PS	37.8947368421052096	37.3684210526316004	46.3157894736842	35.2631578947368	35.7894736842105	38.4210526315789	37.3684210526316004	46.3157894736842	
PTT	57.8947368421052096	55.7894736842105	47.8947368421053	55.7894736842105	55.2631578947368	52.6315789473683396	57.8947368421052096	55.7894736842105	47.3684210526316
PTTEP	53.1578947368421	51.5789473684211105	47.8947368421053	52.6315789473683396	52.6315789473683396	47.3684210526316	51.5789473684211105	51.5789473684211105	47.8947368421053
PTTGC	47.8947368421053	48.4210526315789895	53.6842105263158	50.5263157894737	50.0	57.8947368421052096	48.9473684210526	50.0	53.6842105263158
ROBINS	48.4210526315789895	48.4210526315789895	44.2105263157895	41.5789473684211	44.2105263157895	40.5263157894737	48.4210526315789895	48.4210526315789895	44.2105263157895
SAWAD	53.3333333333333	53.3333333333333	50.0	51.1111111111109	55.5555555555556	52.2222222222222	53.3333333333333	53.3333333333333	48.8888888888889
SCB	60.5263157894737369	57.8947368421052096	41.0526315789474	61.0526315789474105	59.4736842105263	40.0	60.0	57.3684210526316004	41.0526315789474
SCC	47.3684210526316	47.3684210526316	49.4736842105263	51.0526315789474105	49.4736842105263	50.0	47.3684210526316	47.8947368421053	49.4736842105263
TASCO	52.1052631578947	52.1052631578947	47.8947368421053	50.5263157894737	50.5263157894737	47.3684210526316	52.6315789473683396	51.5789473684211105	48.4210526315789895
TCAP	45.2631578947368	45.2631578947368	50.0	45.2631578947368	46.3157894736842	47.3684210526316	45.2631578947368	46.3157894736842	50.0
TMB	52.1052631578947	56.3157894736842	53.6842105263158	52.6315789473683396	56.3157894736842	46.3157894736842	52.1052631578947	55.7894736842105	53.6842105263158
TOP	42.6315789473683396	42.1052631578947104	52.1052631578947	41.5789473684211	45.2631578947368	44.7368421052632	42.6315789473683396	42.1052631578947104	51.0526315789474105
TPIPL	41.5789473684211	42.6315789473683396	56.8421052631579004	45.7894736842105	44.2105263157895	50.0	41.5789473684211	42.6315789473683396	56.3157894736842
TRUE	45.7894736842105	45.2631578947368	50.0	43.6842105263157795	46.8421052631579904	46.3157894736842	45.2631578947368	45.7894736842105	50.0
TTW	48.9473684210526	45.7894736842105	48.4210526315789895	46.8421052631579904	38.9473684210526	48.4210526315789895	48.4210526315789895	45.2631578947368	47.8947368421053
TU	55.2631578947368	56.3157894736842	52.6315789473683396	55.7894736842105	54.7368421052619	52.1052631578947	55.7894736842105	56.3157894736842	51.5789473684211105
WHA	52.1052631578947	46.3157894736842	50.5263157894737	45.2631578947368	45.2631578947368	51.0526315789474105	48.9473684210526	46.8421052631579904	50.5263157894737

Table 103 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	46.842105263157904	50.52631578947437	50.0	44.2105263157895	58.9473684210526	43.684210526315795	46.842105263157904	50.52631578947437
AOT	53.1578947368421	53.1578947368421	54.2105263157895	50.52631578947437	51.052631578947405	55.78947368421055	53.1578947368421	53.1578947368421	54.2105263157895
BA	42.592592592592595	38.88888888888889	40.74074074074074	48.1481481481481	53.703703703703695	57.40740740740741	40.74074074074074	38.88888888888889	40.74074074074074
BANPU	43.1578947368421	46.31578947368421	41.57894736842211	43.1578947368421	45.263157894736	50.52631578947437	43.1578947368421	46.842105263157904	41.57894736842211
BBL	46.842105263157904	46.842105263157904	47.36842105263	45.78947368421055	46.842105263157904	56.31578947368421	46.842105263157904	46.842105263157904	47.36842105263
BCP	42.105263157894704	39.4736842105263	43.15789473368421	42.105263157894704	40.52631578947437	43.15789473368421	42.105263157894704	39.4736842105263	43.15789473368421
BDMS	35.78947368421055	35.78947368421055	43.684210526315795	51.052631578947405	38.94736842105263	55.263157894736	35.78947368421055	35.78947368421055	43.684210526315795
BEC	44.2105263157895	44.2105263157895	46.842105263157904	45.263157894736	44.736842105263	52.1052631578947	44.2105263157895	44.2105263157895	46.842105263157904
BH	56.31578947368421	55.78947368421055	54.2105263157895	52.1052631578947	56.315789473684	56.31578947368421	56.31578947368421	56.31578947368421	54.2105263157895
BLA	45.78947368421055	47.36842105263316	45.263157894736	58.421052631578995	45.263157894736	44.210526315895	45.263157894736	47.36842105263316	45.263157894736
BTS	38.9473684210526	39.47368421052632	42.631578947368396	43.684210526315795	37.368421052631604	48.421052631578895	38.9473684210526	40.52631578947437	42.631578947368396
CBG	34.0	32.0	34.0	38.0	42.0	46.0	32.0	30.0	34.0
CENTEL	53.1578947368421	51.052631578947405	51.578947368421105	45.7894736842105	50.52631578947437	46.31578947368421	53.1578947368421	51.052631578947405	51.578947368421105
CK	38.9473684210526	38.9473684210526	39.47368421052632	42.105263157894704	42.105263157894704	44.736842105263	38.9473684210526	38.9473684210526	39.47368421052632
CPALL	41.57894736842211	42.105263157894704	40.52631578947437	46.842105263157904	46.315789473684	47.89473684210526	41.57894736842211	42.105263157894704	40.52631578947437
CPF	51.578947368421105	53.684210526263158	51.052631578947405	58.421052631578995	54.2105263157895	56.31578947368421	51.578947368421105	53.1578947368421	51.052631578947405
CPN	37.894736842105296	38.9473684210526	43.684210526315795	49.4736842105263	41.578947368421	52.631578947368396	38.4210526315789	38.9473684210526	43.684210526315795
DELTA	50.52631578947437	48.9473684210526	51.052631578947405	51.052631578947405	51.052631578947405	50.52631578947437	50.52631578947437	48.9473684210526	51.052631578947405
DTAC	48.9473684210526	50.0	46.3157894736842	44.7368421052632	47.3684210526316	50.0	48.9473684210526	50.52631578947437	46.3157894736842
EGCO	45.263157894736	48.9473684210526	51.052631578947405	49.4736842105263	46.315789473684	47.3684210526316	45.789473684211	48.421052631578895	51.052631578947405
GLOW	44.7368421052632	46.31578947368421	47.3684210526316	45.7894736842105	45.263157894736	51.578947368421105	44.2105263157895	46.31578947368421	47.36842105263316
GPSC	56.25	43.75	56.25	37.5	62.5	37.5	56.25	43.75	56.25
HMPRO	48.9473684210526	48.421052631578895	41.052631578974	54.2105263157895	56.315789473684	52.631578947368396	48.9473684210526	44.7368421052632	40.52631578947437
INTUCH	51.578947368421105	52.631578947368396	50.52631578947437	55.263157894736	52.1052631578947	46.315789473684	51.578947368421105	52.631578947368396	50.52631578947437
IRPC	47.3684210526316	49.4736842105263	48.421052631578895	48.421052631578895	47.3684210526316	46.315789473684	48.421052631578895	50.52631578947437	48.421052631578895
IVL	42.631578947368396	44.21052631578957	47.36842105263316	46.315789473684	45.263157894736	51.052631578947405	43.15789473684211	43.684210526315795	47.36842105263316
KBANK	48.9473684210526	47.89473684210053	44.7368421052632	45.7894736842105	50.0	47.89473684210053	48.9473684210526	47.89473684210053	44.7368421052632
KCE	54.2105263157895	50.52631578947437	50.0	49.4736842105263	50.0	50.0	53.1578947368421	50.52631578947437	51.052631578947405
KTB	41.57894736842211	40.0	42.105263157894704	44.2105263157895	41.052631578474	48.9473684210526	41.57894736842211	40.0	42.631578947368396
LH	42.105263157894704	41.052631578474	42.631578947368396	51.052631578947405	46.842105263157904	48.421052631578895	42.105263157894704	42.105263157894704	43.684210526315795
MINT	40.52631578947437	41.57894736842211	40.0	52.631578947368396	53.157894736842	51.052631578947405	40.52631578947437	42.631578947368396	40.0
MTLS	61.2244897959184	59.1836734693878	61.2244897959184	57.1428571428571096	57.1428571428571096	48.97959183673467	61.2244897959184	59.1836734693878	61.2244897959184
PS	36.84210526315789	36.31578947368421	37.368421052631604	46.842105263157904	43.157894736842	52.105263157894	36.84210526315789	36.31578947368421	38.94736842105263
PTT	51.578947368421105	56.31578947368421	48.94736842105263	56.842105263157904	55.263157894736	48.421052631578895	52.105263157894	56.31578947368421	49.4736842105263
PTTEP	53.157894736842	52.631578947368396	48.947368421052	53.157894736842	52.631578947368396	52.631578947368396	53.157894736842	52.631578947368396	48.947368421052
PTTGC	47.36842105263316	46.842105263157904	51.052631578947405	53.6842105263158	48.421052631578895	55.263157894736	47.36842105263316	47.36842105263316	51.052631578947405
ROBINS	52.1052631578947	49.473684210263	42.105263157894704	42.105263157894704	48.947368421052	50.0	52.1052631578947	48.947368421052	43.157894736842
SAWAD	52.22222222222222	54.44444444444439	52.22222222222222	42.22222222222222	45.55555555555556	51.11111111111109	53.33333333333333	54.44444444444439	52.22222222222222
SCB	61.052631578947405	61.052631578947405	41.052631578974	59.473684210263	61.052631578947405	60.526315789473694	60.526315789473694	60.526315789473694	41.052631578974
SCC	40.52631578947437	42.105263157894704	42.105263157894704	45.7894736842105	43.157894736842	44.736842105263	40.52631578947437	42.105263157894704	41.57894736842211
TASCO	48.9473684210526	43.684210526315795	48.421052631578895	49.4736842105263	47.89473684210053	45.263157894736	47.89473684210053	43.157894736842	48.421052631578895
TCAP	41.57894736842211	41.57894736842211	40.52631578947437	45.7894736842105	41.052631578974	50.52631578947437	41.57894736842211	41.57894736842211	40.52631578947437
TMB	44.736842105263	43.684210526315795	44.21052631578957	46.315789473684	46.315789473684	45.789473684211	45.263157894736	43.684210526315795	44.736842105263
TOP	44.736842105263	44.2105263157895	43.157894736842	44.736842105263	44.2105263157895	45.263157894736	44.736842105263	44.2105263157895	43.157894736842
TPIPL	40.52631578947437	42.105263157894704	46.315789473684	44.2105263157895	40.52631578947437	54.2105263157895	40.52631578947437	42.105263157894704	46.315789473684
TRUE	42.631578947368396	45.2631578947368	40.52631578947437	44.736842105263	42.105263157894704	47.89473684210053	43.157894736842	45.7894736842105	40.52631578947437
TTW	37.894736842105296	37.894736842105296	47.36842105263316	46.315789473684	40.0	48.9473684210526	37.368421052631604	37.894736842105296	47.36842105263316
TU	54.736842105263194	52.1052631578947	50.0	53.6842105263158	55.789473684210	51.578947368421105	55.263157894736	53.6842105263158	51.052631578947405
WHA	46.842105263157904	46.31578947368421	55.789473684210	54.2105263157895	48.9473684210526	59.4736842105263	46.842105263157904	44.7368421052632	55.789473684210

PredictDirection14 : EMD of tick change

Table 104 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.7368421052632	44.7368421052632	43.684210526315795	45.2631578947368	44.7368421052632	44.2105263157895	44.7368421052632	44.7368421052632	43.684210526315795
AOT	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405	61.052631578947405
BA	55.55555555555556	44.4444444444444	51.851851851851805	50.0	48.1481481481481	51.851851851851805	51.851851851851805	48.1481481481481	53.703703703703695
BANPU	41.052631578974	40.526315789437	41.578947368421	41.052631578974	41.578947368421	41.052631578974	41.052631578974	40.526315789437	41.578947368421
BBL	45.789473684105	46.31578947368421	44.7368421052632	45.789473684105	45.789473684105	45.263157894736	45.789473684105	46.31578947368421	44.7368421052632
BCP	43.157894736842	44.736842105263	45.789473684105	44.2105263157895	42.631578947368396	48.947368421052	43.157894736842	44.736842105263	45.789473684105
BDMS	60.0	58.947368421052	61.578947368421	58.947368421052	58.947368421052	60.0	60.0	58.947368421052	54.2105263157895
BEC	45.263157894736	45.263157894736	45.263157894736	45.263157894736	45.263157894736	45.263157894736	45.263157894736	45.263157894736	45.263157894736
BH	52.631578947368396	53.157894736842	51.052631578947405	52.631578947368396	53.157894736842	51.578947368421105	53.157894736842	53.157894736842	51.052631578947405
BLA	45.789473684105	49.473684210263	45.263157894736	47.894736842105	49.473684210263	47.894736842105	46.842105263157895	48.421052631578895	45.263157894736
BTS	51.052631578947405	44.2105263157895	53.6842105263158	50.52631578947437	46.315789473684	52.631578947368396	53.6842105263158	44.736842105263	53.6842105263158
CBG	70.0	70.0	70.0	70.0	68.0	70.0	70.0	70.0	70.0
CENTEL	58.421052631578995	58.421052631578995	56.842105263157904	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	56.842105263157904
CK	53.6842105263158	58.947368421052	57.368421052631604	56.842105263157904	58.947368421052	57.894736842105296	54.2105263157895	58.947368421052	57.368421052631604
CPALL	51.578947368421105	55.2631578947368	48.421052631578895	52.1052631578947	53.6842105263158	50.52631578947437	52.631578947368396	56.31578947368421	48.421052631578895
CPF	52.631578947368396	53.157894736842	54.736842105263194	52.631578947368396	52.631578947368396	52.631578947368396	52.631578947368396	53.157894736842	54.2105263157895
CPN	53.6842105263158	54.736842105263194	55.2631578947368	54.736842105263194	53.6842105263158	52.631578947368396	53.6842105263158	54.736842105263194	55.7894736842105
DELTA	57.368421052631604	58.421052631578995	53.6842105263158	57.894736842105296	59.473684210263	55.263157894736	56.842105263157904	59.473684210263	54.2105263157895
DTAC	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
EGCO	56.842105263157904	57.894736842105296	56.315789473684	56.842105263157904	57.894736842105296	55.789473684105	57.368421052631604	58.421052631578995	56.315789473684
GLOW	47.36842105263316	48.947368421052	49.4736842105263	52.1052631578947	52.631578947368396	48.947368421052	47.36842105263316	48.947368421052	49.4736842105263
GPSC	43.75	56.25	43.75	43.75	56.25	43.75	43.75	56.25	43.75
HMPRO	52.631578947368396	51.578947368421105	51.052631578947405	52.1052631578947	51.052631578947405	53.6842105263158	53.6842105263158	51.578947368421105	51.052631578947405
INTUCH	43.1578947368421	42.1052631578904	42.105263157894704	42.105263157894704	42.631578947368396	42.105263157894704	43.1578947368421	42.105263157894704	42.105263157894704
IRPC	58.421052631578995	58.947368421052	57.368421052631604	58.947368421052	58.947368421052	57.368421052631604	58.421052631578995	58.947368421052	57.368421052631604
IVL	52.1052631578947	50.0	53.157894736842	51.052631578947405	52.1052631578947	48.947368421052	51.578947368421105	50.0	53.157894736842
KBANK	45.263157894736	44.2105263157895	43.684210526315795	45.263157894736	44.2105263157895	45.263157894736	45.263157894736	43.684210526315795	44.736842105263
KCE	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
KTB	49.4736842105263	46.31578947368421	56.842105263157904	50.52631578947437	46.842105263157904	55.263157894736	48.947368421052	46.842105263157904	56.842105263157904
LH	44.7368421052632	46.842105263157904	42.631578947368396	44.2105263157895	45.263157894736	44.2105263157895	44.2105263157895	46.31578947368421	42.631578947368396
MINT	57.368421052631604	57.368421052631604	57.368421052631604	57.368421052631604	57.368421052631604	57.894736842105296	57.368421052631604	57.368421052631604	57.368421052631604
MTLS	51.020408163265294	51.020408163265294	55.1020408163265	48.9795918367347	51.020408163265294	48.97959183673467	53.061224489795904	51.020408163265294	55.1020408163265
PS	42.631578947368396	49.473684210263	48.947368421052	41.578947368421	49.473684210263	47.894736842105	43.157894736842	49.473684210263	48.947368421052
PTT	43.684210526315795	44.2105263157895	44.2105263157895	44.2105263157895	44.2105263157895	41.578947368421	44.2105263157895	43.684210526315795	43.684210526315795
PTTEP	47.36842105263316	47.89473684210053	46.842105263157904	47.36842105263316	47.89473684210053	47.36842105263316	47.36842105263316	47.89473684210053	46.842105263157904
PTTGC	50.0	50.52631578947437	49.473684210263	50.52631578947437	51.052631578947405	53.6842105263158	50.52631578947437	51.052631578947405	48.421052631578895
ROBINS	53.6842105263158	50.0	53.6842105263158	53.157894736842	49.473684210263	56.315789473684	54.2105263157895	50.0	53.6842105263158
SAWAD	48.88888888888889	48.88888888888889	44.4444444444444	47.77777777778	45.55555555556	41.111111111111	50.0	50.0	44.4444444444444
SCB	40.52631578947437	40.52631578947437	42.631578947368396	40.52631578947437	40.52631578947437	40.52631578947437	40.52631578947437	40.52631578947437	43.157894736842
SCC	48.94736842105296	47.36842105263316	45.2631578947368	48.947368421052	45.789473684105	45.789473684105	48.947368421052	47.89473684210053	45.789473684105
TASCO	44.2105263157895	51.578947368421105	44.2105263157895	44.2105263157895	46.315789473684	43.684210526315795	43.684210526315795	52.1052631578947	44.2105263157895
TCAP	52.631578947368396	54.2105263157895	53.6842105263158	53.157894736842	54.2105263157895	51.578947368421105	52.631578947368396	53.6842105263158	53.6842105263158
TMB	48.947368421052	50.0	51.052631578947405	48.421052631578895	48.421052631578895	51.578947368421105	48.947368421052	50.0	51.052631578947405
TOP	57.894736842105296	59.4736842105263	60.0	57.894736842105296	60.0	60.526315789473694	58.421052631578995	59.4736842105263	60.0
TPIPL	48.947368421052	52.1052631578947	51.052631578947405	48.947368421052	48.421052631578895	53.6842105263158	47.8947368421053	52.631578947368396	51.052631578947405
TRUE	50.52631578947437	48.421052631578895	48.421052631578895	50.52631578947437	48.421052631578895	49.473684210263	50.52631578947437	48.421052631578895	52.1052631578947
TTW	50.0	41.052631578974	57.894736842105296	51.052631578947405	39.473684210263	56.315789473684	50.0	41.052631578974	57.368421052631604
TU	56.31578947368421	54.736842105263194	51.578947368421105	57.368421052631604	53.6842105263158	50.52631578947437	55.789473684210	54.2105263157895	51.578947368421105
WHA	56.31578947368421	50.52631578947437	45.2631578947368	54.736842105263194	51.052631578947405	53.6842105263158	55.263157894736	49.4736842105263	45.2631578947368

Table 105 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	40.52631578947337	41.57894736842211	51.578947368421105	41.05263157894474	42.105263157894704	51.052631578947405	40.52631578947337	41.57894736842211	50.52631578947337
AOT	56.31578947368442	56.31578947368442	52.10526315789474	54.21052631578947	55.78947368421105	50.52631578947337	56.31578947368442	56.31578947368442	52.10526315789474
BA	59.25925925925933	59.25925925925933	53.703703703703695	57.40740740740474	62.962962962963	59.25925925925933	61.11111111111111	59.25925925925933	53.703703703703695
BANPU	57.368421052631604	59.473684210526263	44.736842105263232	57.894736842105296	59.473684210526263	48.947368421052526	56.842105263157904	58.947368421052526	45.26315789473368
BBL	47.89473684210553	46.842105263157904	50.52631578947337	45.26315789473368	48.421052631578895	53.68421052631578	47.89473684210553	46.842105263157904	50.52631578947337
BCP	45.26315789473368	46.31578947368442	47.89473684210553	47.36842105263316	46.842105263157904	51.052631578947405	44.736842105263232	46.842105263157904	47.89473684210553
BDMS	37.368421052631604	36.8421052631579	54.21052631578947	37.368421052631604	35.78947368421105	44.21052631578895	37.368421052631604	36.31578947368442	54.21052631578947
BEC	43.684210526315795	43.684210526315795	52.10526315789474	43.684210526315795	44.736842105263232	48.421052631578895	43.684210526315795	43.684210526315795	52.10526315789474
BH	50.52631578947337	53.68421052631578	52.631578947368396	52.631578947368396	54.21052631578947	53.15789473684211	52.631578947368396	53.68421052631578	53.15789473684211
BLA	44.21052631578895	42.631578947368396	46.31578947368442	41.57894736842211	42.10526315789474	46.842105263157904	44.21052631578895	42.631578947368396	46.842105263157904
BTS	37.368421052631604	34.7368421052632	45.78947368421105	39.473684210526263	37.368421052631604	43.15789473684211	36.84210526315789	34.7368421052632	45.78947368421105
CBG	40.0	42.0	40.0	50.0	46.0	50.0	42.0	42.0	40.0
CENTEL	56.842105263157904	58.421052631578995	52.10526315789474	58.421052631578995	59.473684210526263	48.947368421052526	56.842105263157904	58.421052631578995	52.10526315789474
CK	40.52631578947337	40.0	52.10526315789474	40.0	40.0	44.736842105263232	40.52631578947337	40.0	52.10526315789474
CPALL	45.26315789473368	47.89473684210553	44.73684210526232	46.842105263157904	46.842105263157904	41.57894736842211	45.26315789473368	48.421052631578895	44.21052631578895
CPF	57.894736842105296	56.31578947368442	48.94736842105526	54.73684210526319	56.842105263157904	55.26315789473368	57.894736842105296	55.78947368421105	49.473684210526263
CPN	41.05263157894474	41.57894736842211	41.05263157894474	42.631578947368396	42.105263157894704	44.736842105263232	41.05263157894474	40.52631578947337	42.105263157894704
DELTA	53.15789473684211	53.68421052631578	52.10526315789474	51.052631578947405	52.10526315789474	50.52631578947337	53.15789473684211	53.15789473684211	52.631578947368396
DTAC	48.94736842105526	49.473684210526263	53.15789473684211	49.473684210526263	50.526315789474337	49.473684210526263	48.94736842105526	50.0	53.15789473684211
EGCO	46.842105263157904	53.68421052631578	51.578947368421105	53.15789473684211	56.842105263157904	44.736842105263232	46.31578947368442	53.68421052631578	51.578947368421105
GLOW	46.842105263157904	46.31578947368442	50.0	53.15789473684211	51.052631578947405	53.15789473684211	45.78947368421105	46.31578947368442	50.0
GPSC	56.25	56.25	62.5	43.75	43.75	50.0	56.25	56.25	62.5
HMPRO	61.052631578947405	59.473684210526263	50.0	58.94736842105526	56.31578947368442	50.526315789473737	61.052631578947405	58.421052631578995	50.526315789473737
INTUCH	51.052631578947405	52.631578947368396	53.68421052631578	51.578947368421105	52.10526315789474	52.631578947368396	51.052631578947405	53.15789473684211	54.21052631578947
IRPC	52.10526315789474	52.10526315789474	45.78947368421105	52.10526315789474	52.10526315789474	49.473684210526263	51.052631578947405	52.10526315789474	45.26315789473368
IVL	47.36842105263316	51.578947368421105	54.736842105263219	53.68421052631578	46.842105263157904	51.578947368421105	48.421052631578895	52.10526315789474	54.736842105263219
KBANK	47.36842105263316	48.94736842105526	48.94736842105526	44.21052631578895	45.26315789473368	45.78947368421105	47.36842105263316	49.473684210526263	47.89473684210553
KCE	47.89473684210553	56.842105263157904	50.52631578947337	45.26315789473368	54.21052631578947	50.526315789473737	47.89473684210553	56.31578947368442	50.52631578947337
KTB	42.631578947368396	40.0	41.57894736842211	41.052631578947405	40.0	41.05263157894474	42.631578947368396	40.0	42.105263157894704
LH	46.31578947368442	44.736842105263232	47.36842105263316	50.0	45.263157894736837	44.21052631578895	46.31578947368442	44.736842105263232	47.36842105263316
MINT	43.684210526315795	44.21052631578895	55.78947368421105	43.157894736842211	42.631578947368396	54.736842105263219	43.684210526315795	44.21052631578895	55.78947368421105
MTLS	61.2244897959184	65.30612244897096	65.30612244897096	55.102040816326505	57.142857142857096	61.2244897959184	55.102040816326505	65.30612244897096	65.30612244897096
PS	40.52631578947337	36.84210526315789	53.157894736842101	38.421052631578895	35.789473684210526	55.263157894736837	36.84210526315789	53.157894736842101	53.157894736842101
PTT	57.894736842105296	53.68421052631578	47.36842105263316	57.368421052631604	55.789473684210526	52.10526315789474	57.894736842105296	53.157894736842101	47.89473684210553
PTTEP	51.578947368421105	51.052631578947405	51.052631578947405	48.421052631578895	51.578947368421105	52.631578947368396	51.578947368421105	52.10526315789474	51.578947368421105
PTTGC	48.421052631578895	49.473684210526263	53.68421052631578	50.526315789473737	50.0	57.894736842105296	47.89473684210553	47.89473684210553	53.68421052631578
ROBINS	48.421052631578895	48.421052631578895	44.21052631578895	41.57894736842211	44.21052631578895	40.526315789473737	48.421052631578895	48.421052631578895	44.21052631578895
SAWAD	50.0	52.222222222222	47.77777777777778	55.555555555556	52.222222222222	53.333333333333	50.0	52.222222222222	47.77777777777778
SCB	60.526315789473969	57.368421052631604	41.052631578947474	61.052631578947405	59.473684210526263	40.0	60.0	56.842105263157904	41.052631578947474
SCC	50.526315789473737	49.473684210526263	47.89473684210553	49.473684210526263	48.94736842105526	53.157894736842101	48.947368421052526	50.0	48.421052631578895
TASCO	47.36842105263316	48.94736842105526	48.94736842105526	50.0	45.789473684210526	45.789473684210526	47.368421052631579	47.894736842105263	49.473684210526263
TCAP	45.263157894736837	45.789473684210526	50.0	45.263157894736837	46.315789473684211	47.368421052631579	43.684210526315795	47.368421052631579	50.0
TMB	44.736842105263232	44.21052631578895	44.736842105263232	42.631578947368396	42.631578947368396	43.684210526315795	44.21052631578895	44.21052631578895	44.21052631578895
TOP	42.631578947368396	41.578947368421211	52.105263157894737	41.578947368421211	45.263157894736837	44.736842105263232	42.631578947368396	42.105263157894704	51.052631578947405
TPIPL	44.736842105263232	42.631578947368396	40.526315789473737	48.947368421052526	47.894736842105263	46.842105263157904	44.736842105263232	41.578947368421211	40.526315789473737
TRUE	44.736842105263232	47.894736842105263	45.263157894736368	46.842105263157904	45.789473684210526	46.842105263157904	44.21052631578895	47.368421052631579	45.789473684210526
TTW	41.052631578947474	42.105263157894704	43.157894736842101	37.894736842105296	37.368421052631604	42.105263157894704	43.684210526315795	41.578947368421211	43.157894736842101
TU	53.684210526315795	55.789473684210553	53.157894736842101	55.789473684210553	56.842105263157904	55.789473684210553	54.210526315789495	55.789473684210553	53.684210526315795
WHA	46.31578947368442	44.21052631578895	51.578947368421105	47.368421052631616	44.736842105263232	52.631578947368396	46.31578947368442	44.21052631578895	51.578947368421105

Table 106 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.105263157894704	41.05263157894474	50.0	43.684210526315795	42.105263157894704	46.31578947368442	41.57894736842211	41.05263157894474	49.473684210526263
AOT	57.894736842105296	56.842105263157904	54.21052631578995	55.78947368421105	55.263157894736837	55.263157894736837	57.368421052631604	56.31578947368442	54.736842105263219
BA	59.25925925925933	55.555555555556	57.40740740740474	57.40740740740474	57.40740740740474	62.962962962963	53.703703703703695	57.40740740740474	57.40740740740474
BANPU	57.894736842105296	58.947368421052526	45.263157894736837	57.894736842105296	58.947368421052526	50.526315789473737	57.368421052631604	58.421052631578995	45.789473684210526
BBL	47.36842105263316	46.842105263157904	51.052631578947405	48.947368421052526	48.421052631578895	50.0	47.368421052631616	46.842105263157904	51.052631578947405
BCP	45.263157894736837	46.31578947368442	45.263157894736837	47.894736842105263	47.894736842105263	46.842105263157904	51.052631578947405	46.31578947368442	47.894736842105263
BDMS	37.368421052631604	36.8421052631579	54.21052631578995	37.368421052631604	35.789473684210526	44.21052631578895	37.368421052631604	36.8421052631579	54.21052631578995
BEC	43.157894736842101	43.684210526315795	53.157894736842101	43.684210526315795	44.736842105263232	50.0	43.157894736842101	43.684210526315795	53.157894736842101
BH	53.15789473684211	54.21052631578995	53.684210526315795	52.105263157894737	53.157894736842101	52.105263157894737	52.631578947368396	54.21052631578995	53.684210526315795
BLA	44.21052631578895	42.631578947368396	46.31578947368442	41.578947368421211	42.105263157894704	46.842105263157904	44.21052631578895	42.631578947368396	46.842105263157904
BTS	36.315789473684242	35.263157894736837	45.789473684210526	44.736842105263232	38.947368421052526	47.368421052631616	35.789473684210526	34.736842105263232	45.789473684210526
CBG	40.0	42.0	40.0	50.0	46.0	50.0	42.0	46.0	40.0
CENTEL	57.368421052631604	58.947368421052526	51.578947368421105	58.947368421052526	58.947368421052526	46.842105263157904	56.842105263157904	58.421052631578995	51.578947368421105
CK	40.52631578947337	40.0	49.473684210526263	40.0	40.0	43.684210526315795	40.52631578947337	40.0	49.473684210526263
CPALL	45.263157894736837	47.89473684210553	44.736842105263232	46.842105263157904	46.842105263157904	41.578947368421211	45.263157894736837	47.89473684210553	44.21052631578895
CPF	58.947368421052526	56.31578947368442	51.052631578947405	54.736842105263219	56.842105263157904	48.421052631578895	58.421052631578995	56.842105263157904	51.052631578947405
CPN	42.105263157894704	40.526315789473737	41.052631578947474	42.631578947368396	42.105263157894704	44.736842105263232	42.105263157894704	40.526315789473737	42.105263157894704
DELTA	53.15789473684211	53.684210526315795	52.105263157894737	51.052631578947405	52.105263157894737	50.526315789473737	53.157894736842101	53.157894736842101	52.631578947368396
DTAC	48.947368421052526	50.0	53.157894736842101	47.894736842105263	52.105263157894737	52.631578947368396	48.947368421052526	50.0	53.157894736842101
EGCO	46.315789473684242	52.631578947368396	51.578947368421105	53.157894736842101	56.842105263157904	44.736842105263232	46.315789473684242	53.684210526315795	51.578947368421105
GLOW	46.315789473684242	46.315789473684242	50.0	52.631578947368396	52.105263157894737	53.684210526315795	45.789473684210526	46.315789473684242	49.473684210526263
GPSC	62.5	56.25	62.5	43.75	43.75	50.0	56.25	56.25	62.5
HMPRO	61.052631578947405	59.473684210526263	50.0	58.947368421052526	56.315789473684242	50.526315789473737	61.052631578947405	58.947368421052526	50.526315789473737
INTUCH	51.052631578947405	52.631578947368396	53.684210526315795	51.578947368421105	52.105263157894737	52.631578947368396	51.052631578947405	52.631578947368396	54.210526315789495
IRPC	52.10526315789474	52.10526315789474	44.736842105263232	52.10526315789474	53.157894736842101	49.473684210526263	52.10526315789474	51.578947368421105	44.736842105263232
IVL	51.052631578947405	52.631578947368396	53.157894736842101	54.21052631578995	47.894736842105263	54.210526315789495	52.10526315789474	52.631578947368396	53.157894736842101
KBANK	44.21052631578895	48.421052631578895	48.421052631578895	48.421052631578895	47.368421052631616	45.789473684210526	53.157894736842101	44.736842105263232	48.947368421052526
KCE	48.421052631578895	55.789473684210526	48.947368421052526	45.263157894736837	54.21052631578995	50.526315789473737	48.421052631578895	56.842105263157904	48.947368421052526
KTB	42.105263157894704	40.526315789473737	41.578947368421211	42.631578947368396	39.473684210526263	41.052631578947474	41.052631578947474	40.0	42.105263157894704
LH	48.421052631578895	44.21052631578895	47.36842105263316	46.842105263157904	48.947368421052526	44.21052631578895	47.894736842105263	45.263157894736837	47.36842105263316
MINT	44.21052631578895	44.21052631578895	45.789473684210526	43.157894736842101	43.157894736842101	54.210526315789495	44.21052631578895	44.21052631578895	45.789473684210526
MTLS	61.2244897959184	63.265306122449005	65.30612244897096	53.0612244897959	57.142857142857096	61.2244897959184	57.142857142857096	65.30612244897096	65.30612244897096
PS	38.421052631578	36.84210526315789	55.789473684210526	36.315789473684242	35.263157894736837	57.894736842105296	38.421052631578	36.315789473684242	55.789473684210526
PTT	57.894736842105296	54.21052631578995	47.36842105263316	56.842105263157904	55.789473684210526	52.10526315789474	57.894736842105296	53.157894736842101	47.89473684210553
PTTEP	52.631578947368396	51.052631578947405	48.947368421052526	51.578947368421105	52.631578947368396	52.10526315789474	52.10526315789474	51.052631578947405	47.894736842105263
PTTGC	47.89473684210553	50.526315789473737	53.684210526315795	50.526315789473737	50.526315789473737	57.894736842105296	48.421052631578895	48.947368421052526	53.684210526315795
ROBINS	48.421052631578895	48.421052631578895	44.21052631578895	41.57894736842211	45.789473684210526	40.526315789473737	48.421052631578895	48.421052631578895	44.21052631578895
SAWAD	48.888888888889	52.222222222222	47.77777777777778	55.555555555556	52.222222222222	53.333333333333	50.0	52.222222222222	47.77777777777778
SCB	60.0	57.368421052631604	41.052631578947474	61.052631578947405	59.473684210526263	40.0	60.0	57.368421052631604	41.052631578947474
SCC	49.473684210526263	50.526315789473737	47.894736842105263	48.421052631578895	48.421052631578895	52.105263157894737	48.421052631578895	50.526315789473737	47.894736842105263
TASCO	50.526315789473737	51.052631578947405	48.421052631578895	52.105263157894737	52.105263157894737	50.526315789473737	50.526315789473737	51.052631578947405	47.368421052631616
TCAP	45.263157894736837	45.263157894736837	50.0	45.263157894736837	46.842105263157904	47.368421052631616	45.263157894736837	46.842105263157904	50.0
TMB	44.21052631578895	44.21052631578895	44.210526315789495	42.631578947368396	43.157894736842101	43.157894736842101	44.21052631578895	44.21052631578895	44.21052631578895
TOP	42.105263157894704	41.578947368421211	52.105263157894737	41.578947368421211	46.842105263157904	44.736842105263232	42.631578947368396	42.105263157894704	51.052631578947405
TPIPL	46.842105263157904	42.631578947368396	44.210526315789495	46.842105263157904	45.263157894736837	46.315789473684242	46.842105263157904	43.157894736842101	44.210526315789495
TRUE	44.736842105263232	46.842105263157904	43.157894736842101	46.842105263157904	46.842105263157904	47.368421052631616	44.736842105263232	46.842105263157904	43.157894736842101
TTW	41.578947368421211	43.684210526315795	43.157894736842101	37.368421052631604	36.842105263157879	40.526315789473737	42.631578947368396	43.684210526315795	43.157894736842101
TU	54.736842105263219	55.789473684210553	53.157894736842101	56.315789473684242	56.842105263157904	55.789473684210553	54.210526315789495	55.789473684210553	53.684210526315795
WHA	44.736842105263232	44.210526315789495	53.684210526315795	47.894736842105263	44.736842105263232	52.105263157894737	44.736842105263232	44.210526315789495	53.684210526315795

Table 107 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	41.5789473684211	40.0	46.3157894736842	42.1052631578947	42.1052631578947	42.6315789473684	52.1052631578947	41.5789473684211	40.5263157894737
AOT	54.2105263157895	57.368421052631604	54.736842105263	47.8947368421053	58.421052631578995	46.842105263157904	53.6842105263158	57.368421052631604	54.736842105263
BA	38.8888888888889	38.8888888888889	38.8888888888889	50.0	59.2592592592593	37.037037037037	38.8888888888889	38.8888888888889	38.8888888888889
BANPU	54.2105263157895	57.894736842105296	46.3157894736842	50.0	52.631578947368396	41.052631578947	54.736842105263	57.894736842105296	46.3157894736842
BBL	45.2631578947368	45.7894736842105	50.0	48.421052631578895	46.842105263157904	55.2631578947368	45.2631578947368	45.7894736842105	50.0
BCP	42.1052631578704	40.0	43.1578947368421	42.631578947368396	40.52631578947737	43.1578947368421	42.1052631578704	39.4736842105263	43.1578947368421
BDMS	35.2631578947368	35.2631578947368	41.5789473684211	46.842105263157904	37.894736842105296	57.368421052631604	34.7368421052632	35.2631578947368	41.5789473684211
BEC	43.1578947368421	44.2105263157895	48.421052631578895	47.36842105263158	43.684210526315795	53.684210526315	43.1578947368421	44.2105263157895	48.421052631578895
BH	53.6842105263158	52.1052631578947	51.578947368421105	47.36842105263158	52.1052631578947	55.7894736842105	53.6842105263158	52.1052631578947	51.578947368421105
BLA	45.2631578947368	44.736842105263	44.2105263157895	48.9473684210526	44.2105263157895	52.1052631578947	44.2105263157895	45.2631578947368	44.2105263157895
BTS	37.894736842105296	37.894736842105296	41.5789473684211	41.0526315789474	37.368421052631604	44.736842105263	37.894736842105296	38.4210526315789	42.105263157894704
CBG	34.0	32.0	34.0	38.0	42.0	46.0	34.0	32.0	34.0
CENTEL	56.3157894736842	57.368421052631604	45.7894736842105	52.631578947368396	57.894736842105296	47.3684210526316	55.2631578947368	57.368421052631604	46.3157894736842
CK	38.947368421052	38.947368421052	44.2105263157895	49.473684210263	38.421052631579	47.8947368421053	38.947368421052	38.947368421052	44.2105263157895
CPALL	41.5789473684211	41.5789473684211	40.5263157894737	46.3157894736842	46.3157894736842	47.8947368421053	41.5789473684211	41.0526315789474	40.5263157894737
CPF	53.1578947368421	55.7894736842105	45.7894736842105	54.2105263157895	60.0	47.368421052631	53.1578947368421	56.842105263157904	46.3157894736842
CPN	37.368421052631604	38.947368421052	43.684210526315795	47.8947368421053	41.5789473684211	52.631578947368396	38.947368421052	38.947368421052	43.684210526315795
DELTA	50.5263157894737	48.421052631578895	51.052631578947405	53.684210526315	50.5263157894737	50.5263157894737	51.052631578947405	48.947368421052	51.052631578947405
DTAC	48.421052631578895	48.947368421052	52.631578947368396	47.8947368421053	48.947368421052	49.473684210263	48.947368421052	48.421052631578895	52.631578947368396
EGCO	45.2631578947368	50.0	51.578947368421105	48.421052631578895	46.842105263157904	47.368421052631	44.736842105263	49.473684210263	51.578947368421105
GLOW	48.947368421052	47.3684210526316	45.7894736842105	45.7894736842105	48.947368421052	50.5263157894737	48.421052631578895	48.421052631578895	45.7894736842105
GPSC	56.25	50.0	56.25	43.75	62.5	37.5	56.25	50.0	56.25
HMPRO	58.421052631578995	53.6842105263158	37.894736842105296	55.7894736842105	58.421052631578995	58.947368421052	56.842105263157904	51.052631578947405	37.894736842105296
INTUCH	51.578947368421105	51.578947368421105	50.5263157894737	54.736842105263	51.578947368421105	46.3157894736842	51.052631578947405	52.631578947368396	50.5263157894737
IRPC	48.421052631578895	45.7894736842105	50.5263157894737	51.578947368421105	45.2631578947368	50.5263157894737	48.421052631578895	46.842105263157904	50.5263157894737
IVL	48.947368421052	45.2631578947368	52.1052631578947	45.2631578947368	45.2631578947368	46.3157894736842	48.947368421052	45.2631578947368	52.1052631578947
KBANK	46.3157894736842	50.0	46.3157894736842	44.7368421052632	50.0	44.210526315895	46.842105263157904	50.0	47.368421052631
KCE	45.2631578947368	47.3684210526316	42.105263157894704	50.0	55.7894736842105	51.052631578947405	45.7894736842105	48.421052631578895	41.5789473684211
KTB	40.0	41.0526315789474	41.5789473684211	42.105263157894704	41.0526315789474	40.0	40.0	40.5263157894737	41.5789473684211
LH	42.631578947368396	41.0526315789474	41.0526315789474	45.7894736842105	46.3157894736842	41.5789473684211	42.631578947368396	41.0526315789474	41.0526315789474
MINT	42.631578947368396	43.157894736842	43.684210526315795	49.473684210263	44.736842105263	48.947368421052	42.631578947368396	43.157894736842	44.2105263157895
MTLS	61.2244897959184	59.1836734693878	61.2244897959184	57.1428571428571096	57.1428571428571096	51.020408163265294	61.2244897959184	59.1836734693878	61.2244897959184
PS	36.3157894736842	36.3157894736842	51.052631578947405	54.736842105263	37.368421052631604	54.736842105263	36.3157894736842	36.3157894736842	51.052631578947405
PTT	52.1052631578947	56.3157894736842	46.3157894736842	56.3157894736842	52.631578947368396	48.421052631578895	52.1052631578947	55.7894736842105	47.3684210526316
PTTEP	51.578947368421105	53.157894736842	48.421052631578895	49.473684210263	46.3157894736842	50.0	51.578947368421105	54.210526315895	48.947368421052
PTTGC	46.842105263157904	47.3684210526316	51.052631578947405	54.2105263157895	48.421052631578895	55.2631578947368	47.3684210526316	47.3684210526316	51.052631578947405
ROBINS	52.1052631578947	49.473684210263	42.105263157894704	51.578947368421105	48.947368421052	50.0	52.631578947368396	48.947368421052	43.157894736842
SAWAD	46.6666666666667	51.111111111111109	45.5555555555556	36.6666666666667	47.7777777777778	38.8888888888889	46.6666666666667	51.111111111109	45.5555555555556
SCB	61.052631578947405	61.578947368421	41.052631578974	59.4736842105263	60.526315789473	60.526315789473	60.526315789473	60.526315789473	41.052631578974
SCC	45.7894736842105	45.2631578947368	47.8947368421053	52.631578947368396	49.473684210263	49.473684210263	46.3157894736842	44.736842105263	48.421052631578895
TASCO	48.947368421052	46.842105263157904	52.631578947368396	54.2105263157895	47.3684210526316	50.526315789473	49.473684210263	47.894736842053	52.631578947368396
TCAP	41.5789473684211	41.5789473684211	40.5263157894737	46.842105263157904	41.052631578974	50.526315789473	41.5789473684211	41.5789473684211	40.5263157894737
TMB	42.105263157894704	42.105263157894704	41.052631578974	45.2631578947368	40.0	48.947368421052	42.105263157894704	42.631578947368396	41.052631578974
TOP	44.736842105263	44.2105263157895	43.157894736421	44.2105263157895	44.2105263157895	45.2631578947368	44.736842105263	44.2105263157895	43.157894736421
TPIPL	42.105263157894704	44.2105263157895	41.052631578974	42.631578947368396	43.157894736421	42.105263157894704	41.5789473684211	43.684210526315795	41.052631578974
TRUE	42.631578947368396	45.2631578947368	48.421052631578895	40.5263157894737	44.736842105263	50.526315789473	43.684210526315795	45.2631578947368	48.421052631578895
TTW	35.7894736842105	34.736842105263	34.736842105263	45.2631578947368	41.5789473684211	41.5789473684211	35.7894736842105	34.2105263157895	35.2631578947368
TU	55.2631578947368	52.1052631578947	53.6842105263158	57.368421052631604	51.578947368421105	53.6842105263158	54.736842105263	52.631578947368396	53.6842105263158
WHA	43.1578947368421	44.736842105263	45.2631578947368	53.6842105263158	53.1578947368421	56.3157894736842	42.631578947368396	44.736842105263	46.3157894736842

PredictDirection15 : EMD of direction

Table 108 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.2105263157895	42.631578947368396	44.736842105263	43.1578947368421	43.1578947368421	43.684210526315795	44.2105263157895	42.631578947368396	44.736842105263
AOT	56.842105263157904	58.9473684210526	59.4736842105263	57.368421052631604	58.9473684210526	60.0	56.842105263157904	58.9473684210526	59.4736842105263
BA	44.444444444444	42.592592592592595	44.444444444444	53.703703703703695	53.703703703703695	53.703703703703695	40.7407407407407	42.592592592592595	44.444444444444
BANPU	42.1052631578704	42.631578947368396	42.105263157894704	42.105263157894704	43.1578947368421	42.105263157894704	42.105263157894704	41.5789473684211	42.105263157894704
BBL	41.0526315789474	43.684210526315795	42.105263157894704	40.526315789473737	44.736842105263	43.1578947368421	41.5789473684211	45.7894736842105	42.105263157894704
BCP	52.631578947368396	52.1052631578947	55.2631578947368	46.842105263157904	47.894736842105	52.631578947368396	55.7894736842105	51.578947368421105	55.2631578947368
BDMS	60.0	60.0	59.4736842105263	60.0	60.526315789473737	58.9473684210526	60.0	60.0	0.0
BEC	44.2105263157895	45.2631578947368	45.7894736842105	46.3157894736842	44.736842105263	45.7894736842105	45.2631578947368	45.2631578947368	45.7894736842105
BH	55.7894736842105	54.2105263157895	51.578947368421105	57.368421052631604	55.2631578947368	50.5263157894737	56.842105263157904	55.2631578947368	50.5263157894737
BLA	49.4736842105263	47.894736842105	48.947368421052	50.0	46.842105263157904	49.473684210263	49.4736842105263	49.4736842105263	48.947368421052
BTS	56.3157894736842	54.2105263157895	55.7894736842105	55.7894736842105	49.4736842105263	57.368421052631604	55.7894736842105	54.2105263157895	0.0
CBG	70.0	70.0	68.0	70.0	70.0	68.0	70.0	70.0	68.0
CENTEL	56.842105263157904	54.736842105263	56.842105263157904	56.842105263157904	54.736842105263	55.7894736842105	57.368421052631604	54.736842105263	57.368421052631604
CK	54.2105263157895	55.2631578947368	53.157894736421	53.684210526315	55.263157894736	54.210526315895	54.2105263157895	55.2631578947368	53.157894736421
CPALL	56.3157894736842	55.7894736842105	55.2631578947368	54.2105263157895	53.157894736421	55.2631578947368	56.3157894736842	55.7894736842105	55.2631578947368
CPF	47.8947368421053	50.5263157894737	46.842105263157904	47.8947368421053	49.473684210263	46.842105263157904	47.8947368421053	50.0	47.368421052631
CPN	59.4736842105263	57.894736842105296	59.4736842105263	59.4736842105263	56.842105263157904	56.842105263157904	60.0	57.894736842105296	0.0
DELTA	55.2631578947368	55.7894736842105	54.736842105263	53.684210526315	54.736842105263	57.368421052631604	54.2105263157895	56.842105263157904	54.736842105263
DTAC	49.4736842105263	50.0	50.0	49.473684210263	50.526315789473737	50.0	49.473684210263	50.0	50.0
EGCO	56.3157894736842	56.3157894736842	54.736842105263	56.3157894736842	58.947368421052	56.842105263157904	55.2631578947368	54.2105263157895	54.736842105263
GLOW	50.0	51.052631578947405	56.842105263157904	48.947368421052	48.421052631578895	57.368421052631604	48.947368421052	50.0	56.842105263157904
GPSC	56.25	56.25	56.25	56.25	56.25	50.0	56.25	50.0	56.25
HMPRO	57.894736842105296	57.894736842105296	60.0	58.421052631578995	56.3157894736842	58.421052631578995	57.368421052631604	57.894736842105296	60.0
INTUCH	41.0526315789474	42.105263157894704	41.578947368421	42.105263157894704	42.105263157894704	42.631578947368396	41.0526315789474	42.105263157894704	41.578947368421
IRPC	53.6842105263158	53.6842105263158	53.6842105263158	55.2631578947368	53.157894736421	54.210526315895	55.2631578947368	54.2105263157895	53.6842105263158
IVL	58.947368421052	56.842105263157904	56.842105263157904	56.842105263157904	59.473684210263	58.421052631578995	57.894736842105296	58.421052631578995	56.842105263157904
KBANK	47.3684210526316	47.3684210526316	46.3157894736842	46.842105263157904	46.842105263157904	46.3157894736842	47.3684210526316	47.8947368421053	46.3157894736842
KCE	58.947368421052	61.052631578947405	60.0	60.526315789473737	61.052631578947405	60.0	58.947368421052	61.578947368421	60.0
KTB	51.052631578947405	51.052631578947405	53.6842105263158	49.473684210263	47.368421052631	53.157894736421	49.473684210263	48.947368421052	53.6842105263158
LH	52.631578947368396	54.210526315895	57.368421052631604	44.736842105263	43.684210526315795	53.157894736421	51.052631578947405	54.210526315895	57.368421052631604
MINT	57.894736842105296	57.894736842105296	57.894736842105296	57.894736842105296	57.894736842105296	57.894736842105296	58.421052631578995	57.894736842105296	57.894736842105296
MTLS	51.020408163265294	55.1020408163265	53.061224489759	51.020408163265294	48.979591836734	53.061224489759	53.061224489759	55.1020408163265	53.061224489759
PS	52.1052631578947	56.3157894736842	64.736842105263	51.578947368421105	53.157894736421	65.263157894736	53.157894736421	56.3157894736842	64.736842105263
PTT	40.5263157894737	40.0	43.684210526315795	40.5263157894737	38.947368421052	42.631578947368396	40.5263157894737	41.0526315789474	43.684210526315795
PTTEP	45.2631578947368	47.8947368421053	45.7894736842105	47.368421052631	45.2631578947368	45.7894736842105	45.2631578947368	46.842105263157904	45.7894736842105
PTTGC	55.7894736842105	54.2105263157895	55.2631578947368	56.3157894736842	54.2105263157895	55.2631578947368	55.2631578947368	53.6842105263158	55.2631578947368
ROBINS	49.473684210263	51.578947368421105	53.157894736421	49.473684210263	50.0	50.526315789473	49.473684210263	51.052631578947405	53.157894736421
SAWAD	52.2222222222222	53.3333333333333	46.6666666666667	42.2222222222222	47.7777777777778	43.3333333333333	52.2222222222222	53.3333333333333	46.6666666666667
SCB	43.684210526315795	45.2631578947368	43.157894736421	43.157894736421	42.105263157894704	43.157894736421	44.210526315895	45.7894736842105	42.631578947368396
SCC	46.842105263157904	43.684210526315795	49.473684210263	46.842105263157904	42.105263157894704	47.8947368421053	45.2631578947368	43.684210526315795	48.947368421052
TASCO	51.052631578947405	53.684210526315	49.473684210263	53.684210526315	48.421052631578895	49.473684210263	50.5263157894737	53.684210526315	49.473684210263
TCAP	56.842105263157904	57.368421052631604	55.2631578947368	55.2631578947368	55.2631578947368	53.157894736421	57.368421052631604	55.7894736842105	55.2631578947368
TMB	52.1052631578947	54.2105263157895	57.894736842105296	52.1052631578947	54.2105263157895	55.7894736842105	52.631578947368396	55.2631578947368	57.894736842105296
TOP	58.947368421052	60.0	60.0	60.0	60.526315789473737	59.473684210263	59.473684210263	60.526315789473737	60.0
TPIPL	50.5263157894737	47.894736842105	52.631578947368396	48.947368421052	48.947368421052	52.1052631578947	51.052631578947405	49.473684210263	52.631578947368396
TRUE	47.3684210526316	47.3684210526316	51.578947368421105	49.473684210263	47.3684210526316	51.578947368421105	48.421052631578895	46.842105263157904	51.578947368421105
TTW	46.842105263157904	41.052631578974	62.1052631578947	55.2631578947368	40.0	61.578947368421	46.842105263157904	41.052631578974	62.1052631578947
TU	54.2105263157895	59.4736842105263	55.7894736842105	54.736842105263	55.7894736842105	56.842105263157904	54.736842105263	58.947368421052	55.7894736842105
WHA	53.6842105263158	60.0	56.842105263157904	57.368421052631604	55.7894736842105	56.842105263157904	55.7894736842105	60.0	56.842105263157904

Table 109 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	46.3157894736842	47.3684210526316	44.7368421052632	49.4736842105263	47.3684210526316	50.0	46.3157894736842	45.2631578947368	44.7368421052632
AOT	48.421052631578895	48.947368421052 6	43.684210526315795	38.9473684210526	38.4210526315789	41.5789473684211	48.421052631578895	48.947368421052 6	43.684210526315795
BA	38.8888888888889	38.8888888888889	38.8888888888889	35.185185185185205	38.8888888888889	55.5555555555556	38.8888888888889	38.8888888888889	38.8888888888889
BANPU	45.2631578947368	46.3157894736842	58.421052631578995	48.9473684210526	49.4736842105263	55.2631578947368	48.421052631578895	48.421052631578895	58.421052631578995
BBL	44.7368421052632	42.105263157894704	43.1578947368421	45.2631578947368	48.421052631578895	44.2105263157895	44.7368421052632	42.105263157894704	43.1578947368421
BCP	51.052631578947405	51.052631578947405	48.421052631578895	43.684210526315795	53.1578947368421	42.631578947368396	51.052631578947405	51.052631578947405	48.421052631578895
BDMS	53.6842105263158	54.2105263157895	52.631578947368396	48.421052631578895	45.7894736842105	43.1578947368421	53.6842105263158	53.6842105263158	0.0
BEC	54.2105263157895	54.2105263157895	54.736842105263 19	59.4736842105263	58.9473684210526	48.421052631578895	54.2105263157895	54.2105263157895	54.736842105263 19
BH	43.684210526315795	46.842105263157904	44.7368421052632	48.9473684210526	50.5263157894737	50.0	43.684210526315795	46.3157894736842	44.7368421052632
BLA	53.1578947368421	52.1052631578947	44.2105263157895	51.578947368421105	52.631578947368396	59.4736842105263	51.578947368421105	52.1052631578947	44.2105263157895
BTS	43.684210526315795	46.842105263157904	43.1578947368421	44.7368421052632	41.5789473684211	41.5789473684211	43.684210526315795	46.842105263157904	0.0
CBG	38.0	44.0	38.0	56.00000000000001	52.0	50.0	46.0	44.0	36.0
CENTEL	46.3157894736842	49.4736842105263	42.631578947368396	46.842105263157904	45.7894736842105	41.0526315789474	46.3157894736842	49.4736842105263	42.631578947368396
CK	47.8947368421053	45.2631578947368	49.4736842105263	48.421052631578895	43.684210526315795	50.0	47.8947368421053	45.7894736842105	49.4736842105263
CPALL	53.1578947368421	53.1578947368421	49.4736842105263	48.9473684210526	43.1578947368421	41.5789473684211	52.631578947368396	53.1578947368421	49.4736842105263
CPF	54.736842105263 19	53.6842105263158	56.3157894736842	48.421052631578895	47.3684210526316	50.5263157894737	52.1052631578947	53.6842105263158	55.7894736842105
CPN	57.894736842105296	57.894736842105296	59.4736842105263	38.4210526315789	39.4736842105263	38.9473684210526	57.894736842105296	57.894736842105296	0.0
DELTA	46.3157894736842	48.421052631578895	46.842105263157904	47.8947368421053	44.7368421052632	43.1578947368421	46.3157894736842	45.7894736842105	46.842105263157904
DTAC	50.0	50.526315789473 7	46.3157894736842	51.578947368421105	47.8947368421053	50.5263157894737	50.0	50.526315789473 7	46.3157894736842
EGCO	53.6842105263158	53.1578947368421	43.684210526315795	50.0	51.052631578947405	49.4736842105263	53.6842105263158	53.1578947368421	43.684210526315795
GLOW	50.0	50.0	42.631578947368396	46.3157894736842	47.3684210526316	44.2105263157895	50.0	50.0	42.631578947368396
GPSC	43.75	62.5	43.75	43.75	43.75	43.75	43.75	43.75	43.75
HMPRO	48.9473684210526	49.4736842105263	44.7368421052632	45.7894736842105	47.3684210526316	50.0	48.9473684210526	49.4736842105263	44.7368421052632
INTUCH	56.3157894736842	43.1578947368421	55.7894736842105	55.7894736842105	56.842105263157904	49.4736842105263	56.3157894736842	43.1578947368421	55.7894736842105
IRPC	51.578947368421105	51.052631578947405	53.1578947368421	47.3684210526316	46.3157894736842	45.2631578947368	51.578947368421105	51.052631578947405	53.1578947368421
IVL	54.736842105263 19	54.736842105263 19	57.894736842105296	49.4736842105263	48.421052631578895	54.736842105263 19	51.052631578947405	54.736842105263 19	57.894736842105296
KBANK	43.1578947368421	46.842105263157904	42.105263157894704	50.0	46.842105263157904	46.842105263157904	43.1578947368421	46.842105263157904	42.105263157894704
KCE	51.578947368421105	52.1052631578947	49.4736842105263	44.7368421052632	42.105263157894704	40.0	51.578947368421105	52.1052631578947	49.4736842105263
KTB	48.421052631578895	48.421052631578895	43.684210526315795	45.2631578947368	44.736842105263 2	39.4736842105263	48.421052631578895	48.421052631578895	43.684210526315795
LH	55.2631578947368	55.2631578947368	43.1578947368421	52.631578947368396	53.6842105263158	53.1578947368421	55.2631578947368	55.2631578947368	43.1578947368421
MINT	41.0526315789474	44.2105263157895	40.526315789473 7	47.8947368421053	44.736842105263 2	41.5789473684211	41.0526315789474	44.2105263157895	40.526315789473 7
MTLS	53.0612244897959	53.0612244897959	44.8979591836735	48.9795918367347	48.9795918367347	44.8979591836735	53.0612244897959	53.0612244897959	44.8979591836735
PS	50.526315789473 7	50.0	53.6842105263158	49.4736842105263	48.421052631578895	53.6842105263158	50.526315789473 7	50.0	53.6842105263158
PTT	48.9473684210526	52.631578947368396	51.052631578947405	45.2631578947368	50.5263157894737	42.105263157894704	49.4736842105263	53.1578947368421	50.526315789473 7
PTTEP	54.736842105263 19	48.421052631578895	53.1578947368421	49.4736842105263	47.8947368421053	52.1052631578947	54.736842105263 19	48.421052631578895	53.1578947368421
PTTGC	51.578947368421105	51.052631578947405	52.631578947368396	46.3157894736842	51.052631578947405	54.2105263157895	51.578947368421105	52.1052631578947	52.1052631578947
ROBINS	56.3157894736842	60.0	58.9473684210526	43.1578947368421	47.3684210526316	42.631578947368396	56.3157894736842	60.0	58.9473684210526
SAWAD	60.0	61.111111111111111	60.0	47.7777777777778	42.22222222222222	42.22222222222222	60.0	61.111111111111111	60.0
SCB	47.8947368421053	52.631578947368396	58.421052631578995	58.421052631578995	54.2105263157895	51.052631578947405	51.052631578947405	52.631578947368396	58.421052631578995
SCC	48.421052631578895	48.421052631578895	53.1578947368421	48.9473684210526	48.9473684210526	43.1578947368421	48.421052631578895	48.421052631578895	53.1578947368421
TASCO	51.052631578947405	50.5263157894737	42.105263157894704	46.842105263157904	46.842105263157904	46.3157894736842	51.052631578947405	50.5263157894737	42.105263157894704
TCAP	47.8947368421053	47.8947368421053	44.7368421052632	51.578947368421105	46.842105263157904	46.3157894736842	47.8947368421053	47.8947368421053	44.7368421052632
TMB	48.421052631578895	48.421052631578895	45.7894736842105	46.3157894736842	47.3684210526316	43.684210526315795	48.421052631578895	48.421052631578895	45.7894736842105
TOP	48.421052631578895	44.7368421052632	47.8947368421053	46.3157894736842	45.7894736842105	50.5263157894737	48.421052631578895	44.7368421052632	47.8947368421053
TPIPL	42.631578947368396	42.631578947368396	40.5263157894737	50.0	49.4736842105263	44.2105263157895	37.368421052631604	42.631578947368396	40.5263157894737
TRUE	51.578947368421105	50.5263157894737	46.3157894736842	51.052631578947405	51.578947368421105	52.1052631578947	50.5263157894737	50.5263157894737	46.3157894736842
TTW	42.105263157894704	42.105263157894704	42.105263157894704	39.4736842105263	39.4736842105263	39.4736842105263	42.105263157894704	42.105263157894704	42.105263157894704
TU	43.1578947368421	45.2631578947368	40.5263157894737	43.1578947368421	45.7894736842105	39.4736842105263	43.1578947368421	45.2631578947368	40.5263157894737
WHA	51.052631578947405	49.4736842105263	42.631578947368396	47.3684210526316	50.5263157894737	44.7368421052632	51.052631578947405	49.4736842105263	42.631578947368396

Table 110 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	45.7894736842105	47.3684210526316	45.7894736842105	57.368421052631604	48.9473684210526	48.421052631578895	45.7894736842105	47.3684210526316	45.7894736842105
AOT	55.2631578947368	55.2631578947368	43.684210526315795	44.7368421052632	39.4736842105263	38.9473684210526	42.105263157894704	55.2631578947368	43.684210526315795
BA	50.0	50.0	38.8888888888889	38.8888888888889	38.8888888888889	53.703703703703695	50.0	50.0	38.8888888888889
BANPU	46.842105263157904	46.3157894736842	58.421052631578995	49.4736842105263	49.4736842105263	54.736842105263 19	47.3684210526316	46.3157894736842	58.421052631578995
BBL	42.105263157894704	39.4736842105263	43.1578947368421	46.842105263157904	45.2631578947368	45.2631578947368	42.105263157894704	39.4736842105263	43.1578947368421
BCP	51.052631578947405	51.052631578947405	48.421052631578895	46.842105263157904	42.105263157894704	42.631578947368396	51.052631578947405	51.052631578947405	48.421052631578895
BDMS	53.6842105263158	54.2105263157895	52.631578947368396	49.4736842105263	55.2631578947368	43.1578947368421	53.6842105263158	53.6842105263158	0.0
BEC	53.6842105263158	53.6842105263158	53.6842105263158	58.9473684210526	58.9473684210526	48.9473684210526	53.6842105263158	53.6842105263158	53.6842105263158
BH	46.842105263157904	51.578947368421105	47.8947368421053	51.052631578947405	47.8947368421053	48.9473684210526	46.842105263157904	51.578947368421105	47.8947368421053
BLA	52.1052631578947	52.1052631578947	44.2105263157895	51.578947368421105	52.631578947368396	59.4736842105263	52.1052631578947	52.1052631578947	44.2105263157895
BTS	47.8947368421053	49.4736842105263	43.1578947368421	40.5263157894737	42.105263157894704	41.5789473684211	47.8947368421053	49.4736842105263	0.0
CBG	46.0	44.0	38.0	56.00000000000001	52.0	50.0	46.0	44.0	36.0
CENTEL	49.4736842105263	49.4736842105263	40.0	46.842105263157904	46.842105263157904	41.0526315789474	49.4736842105263	49.4736842105263	40.0
CK	45.2631578947368	44.736842105263 2	47.368421052631 6	46.842105263157904	46.842105263157904	50.526315789473 7	45.2631578947368	44.736842105263 2	49.4736842105263
CPALL	53.1578947368421	53.1578947368421	49.4736842105263	48.9473684210526	43.684210526315795	41.5789473684211	52.631578947368396	53.1578947368421	49.4736842105263
CPF	52.1052631578947	52.1052631578947	56.3157894736842	54.2105263157895	52.1052631578947	48.9473684210526	52.1052631578947	52.1052631578947	56.3157894736842
CPN	57.894736842105296	57.894736842105296	59.4736842105263	38.4210526315789	39.4736842105263	38.9473684210526	57.894736842105296	57.894736842105296	0.0
DELTA	46.3157894736842	45.7894736842105	46.842105263157904	47.8947368421053	44.7368421052632	43.1578947368421	46.3157894736842	45.7894736842105	46.842105263157904
DTAC	51.052631578947405	48.421052631578895	48.421052631578895	50.0	50.5263157894737	52.631578947368396	51.052631578947405	48.9473684210526	48.421052631578895
EGCO	53.6842105263158	53.1578947368421	44.2105263157895	50.0	50.0	49.4736842105263	53.6842105263158	53.1578947368421	44.2105263157895
GLOW	50.0	50.0	42.631578947368396	45.7894736842105	47.3684210526316	44.2105263157895	50.0	50.0	42.631578947368396
GPSC	43.75	62.5	43.75	43.75	43.75	43.75	43.75	62.5	43.75
HMPRO	49.4736842105263	48.9473684210526	44.2105263157895	47.8947368421053	49.4736842105263	50.5263157894737	49.4736842105263	48.421052631578895	44.2105263157895
INTUCH	56.3157894736842	43.1578947368421	55.7894736842105	47.8947368421053	56.3157894736842	49.4736842105263	56.3157894736842	43.1578947368421	55.7894736842105
IRPC	51.578947368421105	51.052631578947405	50.5263157894737	48.421052631578895	45.7894736842105	42.631578947368396	51.578947368421105	51.052631578947405	50.5263157894737
IVL	53.6842105263158	55.7894736842105	44.7368421052632	55.7894736842105	51.052631578947405	48.9473684210526	54.2105263157895	54.736842105263 19	44.736842105263 2
KBANK	47.8947368421053	48.9473684210526	41.5789473684211	44.7368421052632	46.3157894736842	47.8947368421053	47.8947368421053	48.9473684210526	41.5789473684211
KCE	54.2105263157895	50.5263157894737	49.4736842105263	44.7368421052632	43.1578947368421	40.0	52.1052631578947	50.5263157894737	49.4736842105263
KTB	48.421052631578895	48.421052631578895	43.684210526315795	43.684210526315795	44.736842105263 2	39.4736842105263	48.421052631578895	48.421052631578895	43.684210526315795
LH	53.6842105263158	53.6842105263158	56.3157894736842	45.7894736842105	45.2631578947368	41.5789473684211	53.6842105263158	53.6842105263158	56.3157894736842
MINT	42.105263157894704	43.684210526315795	40.526315789473 7	41.5789473684211	45.2631578947368	47.3684210526316	42.105263157894704	43.684210526315795	40.526315789473 7
MTLS	48.9795918367347	53.0612244897959	44.8979591836735	48.9795918367347	48.9795918367347	44.8979591836735	53.0612244897959	53.0612244897959	44.8979591836735
PS	53.1578947368421	52.631578947368396	53.1578947368421	35.7894736842105	35.7894736842105	36.3157894736842	53.1578947368421	52.631578947368396	53.1578947368421
PTT	51.578947368421105	52.631578947368396	51.052631578947405	44.7368421052632	45.2631578947368	42.105263157894704	51.578947368421105	52.631578947368396	50.526315789473 7
PTTEP	52.1052631578947	50.526315789473 7	53.684210526315 8	44.7368421052632	44.7368421052632	52.1052631578947	52.1052631578947	51.052631578947405	53.684210526315 8
PTTGC	51.578947368421105	52.1052631578947	52.631578947368396	46.3157894736842	50.5263157894737	54.2105263157895	51.578947368421105	52.1052631578947	52.1052631578947
ROBINS	56.3157894736842	60.0	58.9473684210526	43.1578947368421	47.3684210526316	42.631578947368396	56.3157894736842	60.0	58.9473684210526
SAWAD	60.0	61.111111111111111	60.0	47.7777777777778	42.22222222222222	42.22222222222222	60.0	61.111111111111111	60.0
SCB	51.578947368421105	52.631578947368396	58.421052631578995	58.421052631578995	54.2105263157895	51.052631578947405	51.578947368421105	52.631578947368396	58.421052631578995
SCC	45.7894736842105	45.7894736842105	51.052631578947405	48.9473684210526	48.9473684210526	43.684210526315795	45.7894736842105	45.7894736842105	51.052631578947405
TASCO	45.7894736842105	46.842105263157904	43.1578947368421	46.842105263157904	47.8947368421053	46.3157894736842	45.2631578947368	46.842105263157904	43.1578947368421
TCAP	47.8947368421053	47.8947368421053	44.7368421052632	51.578947368421105	46.842105263157904	46.3157894736842	47.8947368421053	47.8947368421053	44.7368421052632
TMB	48.421052631578895	48.421052631578895	45.7894736842105	46.3157894736842	47.3684210526316	43.684210526315795	48.421052631578895	48.421052631578895	45.7894736842105
TOP	48.421052631578895	44.7368421052632	47.8947368421053	46.3157894736842	45.7894736842105	50.5263157894737	48.421052631578895	44.7368421052632	47.8947368421053
TPIPL	38.9473684210526	38.9473684210526	40.5263157894737	51.052631578947405	51.052631578947405	44.2105263157895	38.9473684210526	38.9473684210526	40.5263157894737
TRUE	52.631578947368396	44.2105263157895	46.3157894736842	50.0	50.0	52.1052631578947	52.631578947368396	44.736842105263 2	46.3157894736842
TTW	40.5263157894737	40.5263157894737	40.5263157894737	40.5263157894737	38.4210526315789	46.3157894736842	40.5263157894737	40.5263157894737	40.5263157894737
TU	42.105263157894704	45.2631578947368	40.5263157894737	45.2631578947368	45.7894736842105	39.4736842105263	43.1578947368421	45.2631578947368	40.5263157894737
WHA	52.1052631578947	49.4736842105263	42.631578947368396	50.0	48.9473684210526	47.8947368421053	49.4736842105263	49.4736842105263	42.631578947368396

Table 111 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	43.684210526315795	56.31578947368421	51.578947368421105	48.421052631578895	43.684210526315795	43.684210526315795	43.684210526315795
AOT	38.9473684210526	38.9473684210526	38.9473684210526	61.052631578947405	53.6842105263158	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526
BA	38.88888888888889	38.88888888888889	38.88888888888889	44.4444444444444	61.111111111111111	61.111111111111111	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	41.57894736842111	41.57894736842111	41.57894736842111	44.7368421052632	41.57894736842111	44.7368421052632	41.57894736842111	41.57894736842111	41.57894736842111
BBL	44.7368421052632	44.7368421052632	44.7368421052632	55.2631578947368	55.2631578947368	44.2105263157895	44.7368421052632	44.7368421052632	44.7368421052632
BCP	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789	38.4210526315789
BDMS	37.368421052631604	37.368421052631604	37.368421052631604	37.368421052631604	37.368421052631604	52.631578947368396	37.368421052631604	37.368421052631604	0.0
BEC	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368
BH	49.4736842105263	49.4736842105263	49.4736842105263	50.5263157894737	50.5263157894737	50.5263157894737	49.4736842105263	49.4736842105263	49.4736842105263
BLA	45.7894736842105	45.7894736842105	45.7894736842105	56.31578947368421	43.684210526315795	54.2105263157895	45.7894736842105	45.7894736842105	45.7894736842105
BTS	40.52631578947337	40.52631578947337	40.52631578947337	59.4736842105263	56.842105263157904	56.842105263157904	40.52631578947337	40.52631578947337	0.0
CBG	32.0	32.0	32.0	32.0	56.00000000000001	68.0	32.0	32.0	32.0
CENTEL	41.57894736842111	41.57894736842111	41.57894736842111	58.421052631578995	58.421052631578995	40.0	41.57894736842111	41.57894736842111	41.57894736842111
CK	38.9473684210526	38.9473684210526	38.9473684210526	51.578947368421105	48.421052631578895	61.052631578947405	38.9473684210526	38.9473684210526	38.9473684210526
CPALL	43.1578947368421	43.1578947368421	43.1578947368421	51.578947368421105	51.578947368421105	48.421052631578895	43.1578947368421	43.1578947368421	43.1578947368421
CPF	43.684210526315795	43.684210526315795	43.684210526315795	56.31578947368421	56.31578947368421	56.31578947368421	43.684210526315795	43.684210526315795	43.684210526315795
CPN	38.4210526315789	38.4210526315789	38.4210526315789	47.36842105263158	38.4210526315789	52.631578947368396	38.4210526315789	38.4210526315789	0.0
DELTA	45.26315789473368	45.26315789473368	45.26315789473368	46.842105263157904	54.736842105263194	46.842105263157904	45.26315789473368	45.26315789473368	45.26315789473368
DTAC	50.0	50.0	50.0	54.2105263157895	50.0	50.0	50.0	50.0	50.0
EGCO	38.9473684210526	38.9473684210526	38.9473684210526	44.7368421052632	61.052631578947405	44.7368421052632	38.9473684210526	38.9473684210526	38.9473684210526
GLOW	41.57894736842111	41.57894736842111	41.57894736842111	52.1052631578947	52.1052631578947	47.89473684210453	41.57894736842111	41.57894736842111	41.57894736842111
GPSC	43.75	43.75	43.75	43.75	43.75	43.75	43.75	43.75	43.75
HMPRO	36.8421052631579	36.8421052631579	36.8421052631579	63.1578947368421	63.1578947368421	51.578947368421105	36.8421052631579	36.8421052631579	36.8421052631579
INTUCH	42.105263157894704	42.105263157894704	42.105263157894704	57.89473684210596	57.89473684210596	54.2105263157895	42.105263157894704	42.105263157894704	42.105263157894704
IRPC	43.1578947368421	43.1578947368421	43.1578947368421	56.842105263157904	52.1052631578947	47.89473684210453	43.1578947368421	43.1578947368421	43.1578947368421
IVL	42.631578947368396	42.631578947368396	42.631578947368396	57.368421052631604	38.421052631578	52.631578947368396	42.631578947368396	42.631578947368396	42.631578947368396
KBANK	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368	45.26315789473368
KCE	40.0	40.0	40.0	60.0	50.526315789473	49.473684210526	40.0	40.0	40.0
KTB	39.4736842105263	39.4736842105263	39.4736842105263	60.526315789473695	56.31578947368421	43.684210526315795	39.4736842105263	39.4736842105263	39.4736842105263
LH	38.4210526315789	38.4210526315789	38.4210526315789	47.36842105263158	38.4210526315789	47.36842105263158	38.4210526315789	38.4210526315789	38.4210526315789
MINT	42.105263157894704	42.105263157894704	42.105263157894704	47.36842105263158	57.894736842105296	42.105263157894704	42.105263157894704	42.105263157894704	42.105263157894704
MTLS	48.97959183673347	48.97959183673347	48.97959183673347	48.97959183673347	48.97959183673347	44.89795918367345	48.97959183673347	48.97959183673347	48.97959183673347
PS	36.31578947368342	36.31578947368342	36.31578947368342	46.31578947368421	63.684210526315795	36.31578947368342	36.31578947368342	36.31578947368342	36.31578947368342
PTT	41.05263157894474	41.05263157894474	41.05263157894474	58.947368421052596	58.947368421052596	58.947368421052596	41.05263157894474	41.05263157894474	41.05263157894474
PTTEP	54.2105263157895	54.2105263157895	54.2105263157895	52.631578947368396	47.36842105263158	54.2105263157895	54.2105263157895	54.2105263157895	54.2105263157895
PTTGC	46.842105263157904	46.842105263157904	46.842105263157904	53.157894736842105	53.157894736842105	46.842105263157904	46.842105263157904	46.842105263157904	46.842105263157904
ROBINS	37.894736842105296	37.894736842105296	37.894736842105296	62.1052631578947	62.1052631578947	62.1052631578947	37.894736842105296	37.894736842105296	37.894736842105296
SAWAD	41.11111111111111	41.11111111111111	41.11111111111111	55.55555555555556	55.55555555555556	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111
SCB	41.05263157894474	41.05263157894474	41.05263157894474	48.421052631578895	58.947368421052596	48.421052631578895	41.05263157894474	41.05263157894474	41.05263157894474
SCC	40.0	40.0	40.0	46.842105263157904	46.842105263157904	40.0	40.0	40.0	40.0
TASCO	46.31578947368342	46.31578947368342	46.31578947368342	53.6842105263158	46.31578947368342	44.7368421052632	46.31578947368342	46.31578947368342	46.31578947368342
TCAP	38.4210526315789	38.4210526315789	38.4210526315789	61.5789473684211	61.5789473684211	61.5789473684211	38.4210526315789	38.4210526315789	38.4210526315789
TMB	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795
TOP	40.0	40.0	40.0	52.1052631578947	60.0	47.89473684210453	40.0	40.0	40.0
TPIPL	44.2105263157895	44.2105263157895	44.2105263157895	50.526315789473695	50.526315789473695	54.4736842105263	44.2105263157895	44.2105263157895	44.2105263157895
TRUE	45.26315789473368	45.26315789473368	45.26315789473368	46.31578947368421	53.6842105263158	54.736842105263194	45.26315789473368	45.26315789473368	45.26315789473368
TTW	34.7368421052632	34.7368421052632	34.7368421052632	47.89473684210453	65.26315789473368	65.26315789473368	34.7368421052632	34.7368421052632	34.7368421052632
TU	40.0	40.0	40.0	47.36842105263158	40.0	40.0	40.0	40.0	40.0
WHA	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632	44.7368421052632

PredictDirection16 : MACD

Table 112 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.631578947368396	43.157894736842105	44.21052631578947	42.631578947368396	43.684210526315795	43.684210526315795	43.157894736842105	43.684210526315795	44.21052631578947
AOT	60.526315789473695	61.052631578947405	61.052631578947405	60.526315789473695	61.052631578947405	61.052631578947405	60.526315789473695	61.052631578947405	61.052631578947405
BA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	48.148148148148145	48.148148148148145
BANPU	43.684210526315795	42.631578947368396	42.631578947368396	42.631578947368396	41.052631578947405	42.631578947368396	43.684210526315795	42.631578947368396	42.631578947368396
BBL	44.736842105263194	47.894736842105263	43.157894736842105	43.684210526315795	44.736842105263194	44.736842105263194	45.789473684210526	44.21052631578947	43.684210526315795
BCP	53.684210526315795	56.842105263157904	54.21052631578947	53.684210526315795	49.473684210526316	52.105263157894704	56.842105263157904	56.842105263157904	54.736842105263194
BDMS	58.421052631578995	60.0	58.947368421052596	59.473684210526316	60.0	59.473684210526316	58.421052631578995	60.0	58.947368421052596
BEC	45.789473684210526	44.21052631578947	45.789473684210526	44.21052631578947	44.21052631578947	47.36842105263158	45.789473684210526	45.26315789473368	45.789473684210526
BH	48.94736842105263	48.94736842105263	49.473684210526316	51.578947368421105	50.526315789473695	48.94736842105263	48.94736842105263	50.0	49.473684210526316
BLA	46.842105263157904	46.842105263157904	50.0	47.36842105263158	45.26315789473368	48.94736842105263	46.842105263157904	46.842105263157904	49.473684210526316
BTS	48.94736842105263	47.36842105263158	56.842105263157904	52.1052631578947	47.894736842105353	54.736842105263194	47.89473684210453	46.842105263157904	56.842105263157904
CBG	68.0	68.0	68.0	68.0	68.0	68.0	68.0	68.0	68.0
CENTEL	57.894736842105296	57.368421052631604	58.94736842105263	56.842105263157904	58.421052631578995	58.421052631578995	57.894736842105296	57.368421052631604	58.94736842105263
CK	54.2105263157895	54.736842105263194	57.368421052631604	55.26315789473368	55.26315789473368	55.789473684210526	54.736842105263194	54.2105263157895	58.421052631578995
CPALL	52.1052631578947	48.94736842105263	53.157894736842105	51.052631578947405	50.0	52.631578947368396	52.631578947368396	48.94736842105263	53.157894736842105
CPF	45.26315789473368	45.26315789473368	46.842105263157904	46.842105263157904	45.26315789473368	46.842105263157904	45.26315789473368	47.368421052631604	46.842105263157904
CPN	53.684210526315795	55.26315789473368	54.736842105263194	56.842105263157904	56.842105263157904	56.31578947368342	54.2105263157895	55.26315789473368	54.736842105263194
DELTA	57.368421052631604	52.631578947368396	55.789473684210526	56.31578947368342	55.789473684210526	57.368421052631604	56.31578947368342	55.789473684210526	55.26315789473368
DTAC	50.0	50.0	48.947368421052596	50.0	50.526315789473695	48.947368421052596	50.0	50.0	48.947368421052596
EGCO	58.421052631578995	57.368421052631604	57.894736842105296	57.368421052631604	57.368421052631604	58.421052631578995	57.894736842105296	56.842105263157904	57.368421052631604
GLOW	48.94736842105263	54.2105263157895	54.736842105263194	49.473684210526316	46.31578947368342	51.578947368421105	49.473684210526316	53.157894736842105	54.736842105263194
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	60.526315789473695	57.368421052631604	60.526315789473695	59.473684210526316	56.31578947368342	60.526315789473695	59.473684210526316	57.368421052631604	60.526315789473695
INTUCH	41.57894736842111	42.631578947368396	41.05263157894474	41.57894736842111	42.631578947368396	41.05263157894474	41.57894736842111	42.105263157894704	41.05263157894474
IRPC	55.26315789473368	54.736842105263194	53.157894736842105	53.157894736842105	55.789473684210526	53.157894736842105	54.736842105263194	54.736842105263194	53.157894736842105
IVL	56.842105263157904	53.157894736842105	54.736842105263194	54.736842105263194	53.684210526315795	55.263157894736842	57.368421052631604	52.105263157894704	56.31578947368342
KBANK	47.89473684210453	48.421052631578895	44.7368421052632	47.89473684210453	44.736842105263194	47.36842105263158	46.31578947368342	45.789473684210526	44.736842105263194
KCE	59.473684210526316	60.526315789473695	60.526315789473695	58.421052631578995	61.052631578947405	60.526315789473695	58.947368421052596	60.526315789473695	60.526315789473695
KTB	55.26315789473368	48.421052631578895	58.947368421052596	51.052631578947405	39.473684210526316	54.2105263157895	54.736842105263194	49.473684210526316	58.947368421052596
LH	42.105263157894704	41.052631578947405	50.526315789473695	45.263157894736842	39.473684210526316	51.578947368421105	43.157894736842105	40.0	50.526315789473695
MINT	58.421052631578995	57.894736842105296	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	58.421052631578995	57.894736842105296	58.421052631578995
MTLS	0.0	0.0	0.0	0.0	0.0	0.0	55.1020408163265	57.142857142857096	57.142857142857096
PS	56.31578947368342	56.842105263157904	65.78947368421049	53.6842105263158	44.736842105263194	65.26315789473368	62.1052631578947	47.894736842105353	65.78947368421049
PTT	46.31578947368342	43.157894736842105	43.157894736842105	43.684210526315795	44.736842105263194	42.631578947368396	45.789473684210526	43.684210526315795	43.157894736842105
PTTEP	48.421052631578895	45.789473684210526	42.105263157894704	50.0	47.368421052631604	42.105263157894704	47.894736842105353	46.842105263157904	42.105263157894704
PTTGC	55.26315789473368	46.31578947368342	55.789473684210526	50.526315789473695	47.894736842105353	53.157894736842105	54.2105263157895	46.31578947368342	55.789473684210526
ROBINS	52.1052631578947	51.578947368421105	50.0	48.947368421052596	53.157894736842105	50.526315789473695	50.526315789473695	53.157894736842105	50.0
SAWAD	41.11111111111111	43.333333333333336	42.22222222222222	42.22222222222222	44.44444444444444	41.11111111111111	40.0	42.22222222222222	42.22222222222222
SCB	42.105263157894704	44.21052631578947	42.631578947368396	42.105263157894704	41.578947368421105	41.05263157894474	42.631578947368396	44.736842105263194	42.631578947368396
SCC	46.31578947368342	44.736842105263194	47.894736842105353	47.894736842105353	46.315789473684205	49.473684210526316	47.368421052631604	46.31578947368342	47.894736842105353
TASCO	53.684210526315795	53.684210526315795	53.684210526315795	53.684210526315795	53.684210526315795	53.684210526315795	53.684210526315795	53.684210526315795	53.684210526315795
TCAP	56.842105263157904	55.789473684210526	54.21052631578947	60.0	55.789473684210526	53.684210526315795	55.789473684210526	55.789473684210526	54.21052631578947
TMB	47.368421052631604	48.421052631578895	57.368421052631604	51.052631578947405	45.789473684210526	58.947368421052596	47.368421052631604	49.473684210526316	58.947368421052596
TOP	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
TPIPL	54.2105263157895	47.368421052631604	52.1052631578947	57.894736842105296	55.789473684210526	54.736842105263194	54.736842105263194	48.421052631578895	51.578947368421105
TRUE	47.894736842105353	45.789473684210526	52.631578947368396	46.842105263157904	44.736842105263194	52.631578947368396	46.31578947368342	46.842105263157904	53.157894736842105
TTW	41.052631578947405	44.21052631578947	60.526315789473695	38.421052631578995	40.0	59.473684210526316	39.473684210526316	40.526315789473695	60.526315789473695
TU	56.31578947368342	53.157894736842105	55.26315789473368	54.2105263157895	52.631578947368396	56.842105263157904	56.31578947368342	52.1052631578947	55.26315789473368
WHA	58.947368421052596	55.789473684210526	58.421052631578995	51.052631578947405	57.368421052631604	56.842105263157904	56.842105263157904	55.26315789473368	61.052631578947405

Table 113 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	50.0	49.4736842105263	43.684210526315795	44.7368421052632	46.3157894736842	45.7894736842105	50.0	49.4736842105263	43.684210526315795
AOT	37.368421052631604	43.684210526315795	39.4736842105263	41.5789473684211	41.5789473684211	42.631578947368396	40.52631578947237	43.684210526315795	39.4736842105263
BA	0.0	0.0	0.0	0.0	0.0	0.0	44.44444444444444	44.44444444444444	38.88888888888889
BANPU	44.7368421052632	44.2105263157895	58.9473684210526	43.684210526315795	44.7368421052632	42.105263157894704	44.2105263157895	41.0526315789474	58.9473684210526
BBL	41.0526315789474	41.5789473684211	44.7368421052632	43.1578947368421	47.8947368421053	42.631578947368396	41.0526315789474	41.5789473684211	44.7368421052632
BCP	42.631578947368396	44.7368421052632	38.4210526315789	42.631578947368396	41.0526315789474	38.4210526315789	42.631578947368396	44.7368421052632	38.4210526315789
BDMS	54.2105263157895	54.73684210526319	60.0	55.2631578947368	40.52631578947237	40.0	54.2105263157895	54.2105263157895	60.0
BEC	51.578947368421105	53.1578947368421	52.10526315578947	46.842105263157904	46.842105263157904	43.684210526315795	51.578947368421105	53.1578947368421	52.10526315578947
BH	55.7894736842105	50.0	50.52631578947237	55.2631578947368	55.2631578947368	52.10526315578947	58.421052631578995	50.0	50.0
BLA	51.052631578947405	52.631578947368396	54.2105263157895	50.0	50.0	56.3157894736842	50.0	52.631578947368396	54.2105263157895
BTS	53.1578947368421	55.7894736842105	56.3157894736842	51.578947368421105	54.2105263157895	47.89473684210053	53.1578947368421	55.7894736842105	54.73684210526319
CBG	32.0	32.0	40.0	32.0	32.0	32.0	32.0	36.0	32.0
CENTEL	50.0	51.052631578947405	41.5789473684211	54.2105263157895	54.2105263157895	41.5789473684211	50.0	51.052631578947405	41.5789473684211
CK	38.9473684210526	46.3157894736842	38.9473684210526	51.052631578947405	51.052631578947405	43.684210526315795	39.4736842105263	46.842105263157904	38.4210526315789
CPALL	54.2105263157895	55.7894736842105	48.421052631578895	50.52631578947237	50.52631578947237	41.5789473684211	54.2105263157895	55.7894736842105	48.421052631578895
CPF	47.89473684210053	53.1578947368421	54.73684210526319	52.631578947368396	55.7894736842105	52.631578947368396	50.52631578947237	53.6842105263158	54.73684210526319
CPN	51.052631578947405	51.578947368421105	35.7894736842105	40.0	47.36842105263316	40.52631578947237	51.052631578947405	51.578947368421105	35.26315789473368
DELTA	50.52631578947237	48.94736842105526	46.842105263157904	48.9473684210526	55.2631578947368	46.3157894736842	50.0	48.94736842105526	46.842105263157904
DTAC	47.36842105263316	48.94736842105526	53.1578947368421	47.36842105263316	47.36842105263316	47.89473684210053	48.94736842105526	48.94736842105526	52.631578947368396
EGCO	52.631578947368396	48.421052631578895	42.105263157894704	46.842105263157904	44.7368421052632	52.631578947368396	52.631578947368396	48.421052631578895	42.105263157894704
GLOW	46.842105263157904	44.736842105263216	41.5789473684211	43.1578947368421	41.052631578947	41.052631578947	43.1578947368421	44.736842105263216	41.578947368421
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	46.842105263157904	45.7894736842105	43.1578947368421	44.7368421052632	43.1578947368421	43.1578947368421	46.842105263157904	42.631578947368396	43.1578947368421
INTUCH	43.1578947368421	43.1578947368421	42.105263157894704	37.894736842105296	40.0	42.105263157894704	42.631578947368396	43.1578947368421	42.105263157894704
IRPC	46.842105263157904	48.421052631578895	43.1578947368421	44.2105263157895	43.684210526315795	43.1578947368421	46.842105263157904	48.421052631578895	43.1578947368421
IVL	56.3157894736842	54.73684210526319	53.1578947368421	51.578947368421105	54.73684210526319	53.6842105263158	50.52631578947237	54.73684210526319	54.2105263157895
KBANK	45.7894736842105	43.684210526315795	45.263157894736842	44.2105263157895	49.473684210263	48.421052631578895	47.368421052636316	46.315789473684	45.263157894736842
KCE	47.89473684210053	49.4736842105263	49.4736842105263	54.73684210526319	47.89473684210053	45.7894736842105	47.89473684210053	50.52631578947237	49.4736842105263
KTB	42.105263157894704	42.105263157894704	40.52631578947237	40.52631578947237	48.421052631578895	39.473684210263	42.105263157894704	42.105263157894704	40.52631578947237
LH	53.1578947368421	52.1052631578947	38.4210526315789	43.684210526315795	44.2105263157895	37.894736842105296	51.578947368421105	52.1052631578947	38.4210526315789
MINT	43.684210526315795	46.315789473684242	42.105263157894704	50.0	50.0	54.2105263157895	43.684210526315795	44.736842105263216	42.105263157894704
MTLS	0.0	0.0	0.0	0.0	0.0	0.0	46.9387755102041	53.061244897959	44.8979591836735
PS	42.105263157894704	41.5789473684211	36.3157894736842	48.421052631578895	49.473684210263	41.5789473684211	40.52631578947237	41.5789473684211	36.3157894736842
PTT	55.7894736842105	58.9473684210526	58.9473684210526	51.578947368421105	53.6842105263158	42.631578947368396	52.631578947368396	58.9473684210526	58.9473684210526
PTTEP	47.89473684210053	53.1578947368421	52.631578947368396	52.10526315787947	48.421052631578895	52.10526315787947	52.10526315787947	53.1578947368421	52.631578947368396
PTTGC	54.2105263157895	54.2105263157895	52.631578947368396	56.842105263157904	55.2631578947368	52.10526315787947	54.2105263157895	53.6842105263158	48.9473684210526
ROBINS	54.73684210526319	55.7894736842105	40.52631578947237	43.1578947368421	43.1578947368421	37.894736842105296	54.73684210526319	55.7894736842105	41.5789473684211
SAWAD	42.22222222222222	42.22222222222222	41.11111111111111	50.0	51.111111111111109	55.555555555556	42.22222222222222	43.333333333333	41.11111111111111
SCB	55.263157894736842	48.9473684210526	48.9473684210526	52.631578947368396	54.2105263157895	41.052631578947	51.578947368421105	55.7894736842105	51.052631578947405
SCC	45.263157894736842	46.842105263157904	40.0	47.36842105263316	51.052631578947405	43.684210526315795	45.263157894736842	46.842105263157904	40.0
TASCO	47.36842105263316	44.736842105263216	50.0	48.9473684210526	48.9473684210526	46.3157894736842	47.36842105263316	44.736842105263216	50.0
TCAP	53.1578947368421	51.578947368421105	46.3157894736842	47.36842105263316	47.36842105263316	49.473684210263	53.1578947368421	51.052631578947405	46.3157894736842
TMB	54.73684210526319	54.2105263157895	40.0	48.421052631578895	48.421052631578895	48.947368421052	54.73684210526319	54.2105263157895	40.0
TOP	44.2105263157895	44.736842105263216	50.0	46.3157894736842	51.578947368421105	53.6842105263158	44.2105263157895	44.736842105263216	50.0
TPIPL	43.684210526315795	46.3157894736842	44.2105263157895	61.052631578947405	57.894736842105296	65.263157894736	45.263157894736	44.2105263157895	44.2105263157895
TRUE	52.631578947368396	53.1578947368421	45.263157894736	56.3157894736842	53.1578947368421	45.263157894736	52.631578947368396	53.1578947368421	45.263157894736
TTW	45.263157894736	45.263157894736	42.105263157894704	41.5789473684211	42.631578947368396	37.894736842105296	44.2105263157895	45.263157894736	42.105263157894704
TU	47.89473684210053	55.263157894736	45.263157894736	53.1578947368421	56.842105263157904	61.052631578947405	49.473684210263	55.263157894736	45.263157894736
WHA	51.578947368421105	52.1052631578947	44.7368421052632	48.9473684210526	50.52631578947237	50.52631578947237	52.1052631578947	52.1052631578947	44.7368421052632

Table 114 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	51.052631578947405	51.052631578947405	43.684210526315795	45.7894736842105	52.10526315787947	47.3684210526316	51.052631578947405	51.052631578947405	43.684210526315795
AOT	41.5789473684211	43.15789473684211	38.9473684210526	38.9473684210526	41.052631578947	42.105263157894704	41.5789473684211	43.15789473684211	38.9473684210526
BA	0.0	0.0	0.0	0.0	0.0	0.0	55.5555555555556	35.185185185185205	38.88888888888889
BANPU	44.7368421052632	41.0526315789474	58.9473684210526	44.2105263157895	45.263157894736842	42.105263157894704	44.2105263157895	41.0526315789474	58.9473684210526
BBL	42.631578947368396	42.631578947368396	44.7368421052632	42.631578947368396	47.8947368421053	42.631578947368396	42.105263157894704	42.631578947368396	45.263157894736842
BCP	42.631578947368396	44.736842105263216	38.4210526315789	42.631578947368396	41.052631578947	38.4210526315789	42.631578947368396	44.736842105263216	38.4210526315789
BDMS	54.2105263157895	54.2105263157895	60.0	47.8947368421053	40.52631578947237	40.0	54.2105263157895	54.2105263157895	60.0
BEC	47.36842105263316	48.421052631578895	52.631578947368396	46.842105263157904	46.842105263157904	45.263157894736842	47.368421052636316	48.421052631578895	52.631578947368396
BH	54.73684210526319	47.36842105263316	49.4736842105263	51.052631578947405	56.315789473684242	50.52631578947237	55.263157894736842	47.8947368421053	48.9473684210526
BLA	50.0	52.631578947368396	54.2105263157895	55.7894736842105	50.0	56.3157894736842	51.052631578947405	52.631578947368396	54.2105263157895
BTS	53.1578947368421	55.7894736842105	54.73684210526319	48.421052631578895	54.2105263157895	47.89473684210053	53.1578947368421	55.7894736842105	54.73684210526319
CBG	32.0	36.0	40.0	32.0	32.0	32.0	32.0	36.0	32.0
CENTEL	54.73684210526319	51.052631578947405	41.5789473684211	49.4736842105263	51.578947368421105	41.5789473684211	54.73684210526319	51.052631578947405	41.5789473684211
CK	42.631578947368396	41.5789473684211	38.9473684210526	54.73684210526319	46.315789473684242	53.1578947368421	42.105263157894704	41.5789473684211	38.9473684210526
CPALL	54.2105263157895	55.7894736842105	48.421052631578895	50.52631578947237	50.52631578947237	41.5789473684211	54.2105263157895	55.7894736842105	48.421052631578895
CPF	50.52631578947237	51.052631578947405	52.631578947368396	50.52631578947237	51.578947368421105	53.6842105263158	50.52631578947237	51.052631578947405	53.6842105263158
CPN	51.052631578947405	51.578947368421105	35.7894736842105	40.0	47.36842105263316	40.52631578947237	51.052631578947405	51.578947368421105	35.26315789473368
DELTA	50.52631578947237	48.94736842105526	46.842105263157904	54.2105263157895	55.2631578947368	46.3157894736842	50.52631578947237	48.94736842105526	46.842105263157904
DTAC	47.89473684210053	46.842105263157904	52.631578947368396	48.421052631578895	48.947368421052	55.263157894736	50.0	46.842105263157904	52.631578947368396
EGCO	53.1578947368421	48.94736842105526	42.631578947368396	47.36842105263316	45.2631578947368	52.10526315787947	53.1578947368421	48.94736842105526	42.631578947368396
GLOW	46.842105263157904	41.5789473684211	41.5789473684211	43.1578947368421	41.052631578947	41.052631578947	43.1578947368421	43.1578947368421	41.5789473684211
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	47.36842105263316	45.263157894736842	43.684210526315795	47.36842105263316	46.315789473684242	43.684210526315795	47.36842105263316	46.315789473684242	43.684210526315795
INTUCH	43.684210526315795	43.1578947368421	42.105263157894704	37.894736842105296	40.0	42.105263157894704	43.684210526315795	43.1578947368421	42.105263157894704
IRPC	51.052631578947405	51.578947368421105	43.1578947368421	45.7894736842105	46.315789473684242	43.1578947368421	51.052631578947405	51.578947368421105	43.1578947368421
IVL	55.7894736842105	56.3157894736842	61.5789473684211	48.9473684210526	51.052631578947405	47.36842105263316	54.73684210526319	56.3157894736842	61.5789473684211
KBANK	51.578947368421105	51.578947368421105	44.736842105263216	49.473684210263	50.0	48.947368421052	50.52631578947237	51.578947368421105	44.736842105263216
KCE	51.052631578947405	51.052631578947405	50.52631578947237	52.631578947368396	54.73684210526319	40.0	51.052631578947405	48.421052631578895	49.4736842105263
KTB	42.105263157894704	42.105263157894704	40.52631578947237	40.52631578947237	40.52631578947237	39.473684210263	42.105263157894704	42.105263157894704	40.52631578947237
LH	51.052631578947405	54.73684210526319	37.368421052631604	42.631578947368396	38.4210526315789	37.894736842105296	51.052631578947405	54.73684210526319	37.368421052631604
MINT	44.736842105263216	46.315789473684242	42.105263157894704	50.52631578947237	50.0	47.89473684210053	44.736842105263216	46.315789473684242	42.105263157894704
MTLS	0.0	0.0	0.0	0.0	0.0	0.0	46.9387755102041	53.061244897959	44.8979591836735
PS	53.1578947368421	39.473684210263	38.9473684210526	50.52631578947237	39.473684210263	60.0	53.1578947368421	39.473684210263	38.9473684210526
PTT	54.2105263157895	58.9473684210526	58.9473684210526	50.52631578947237	46.842105263157904	42.631578947368396	56.3157894736842	58.9473684210526	58.9473684210526
PTTEP	51.052631578947405	50.52631578947237	55.263157894736842	45.7894736842105	51.578947368421105	55.789473684210	51.578947368421105	52.10526315787947	53.6842105263158
PTTGC	55.7894736842105	54.2105263157895	52.631578947368396	56.842105263157904	55.7894736842105	52.10526315787947	54.2105263157895	54.736842105263216	48.9473684210526
ROBINS	54.73684210526319	55.7894736842105	40.52631578947237	43.1578947368421	43.1578947368421	37.894736842105296	45.263157894736	55.7894736842105	41.5789473684211
SAWAD	42.22222222222222	43.333333333333	41.11111111111111	50.0	50.0	55.555555555556	42.22222222222222	43.333333333333	41.11111111111111
SCB	55.263157894736	55.7894736842105	48.9473684210526	50.52631578947237	54.2105263157895	41.052631578947	55.263157894736	55.7894736842105	51.052631578947405
SCC	44.2105263157895	45.7894736842105	40.0	44.2105263157895	48.421052631578895	45.7894736842105	44.2105263157895	45.7894736842105	40.0
TASCO	43.684210526315795	44.736842105263216	42.631578947368396	46.315789473684242	45.263157894736	42.631578947368396	43.684210526315795	43.684210526315795	42.631578947368396
TCAP	53.1578947368421	51.578947368421105	46.3157894736842	47.36842105263316	47.36842105263316	49.473684210263	53.1578947368421	51.052631578947405	46.3157894736842
TMB	55.7894736842105	55.7894736842105	43.684210526315795	48.421052631578895	48.421052631578895	46.842105263157904	55.7894736842105	55.7894736842105	43.684210526315795
TOP	44.2105263157895	44.736842105263216	50.0	46.3157894736842	51.578947368421105	53.6842105263158	44.2105263157895	44.736842105263216	50.0
TPIPL	42.631578947368396	41.5789473684211	44.2105263157895	62.631578947368	62.631578947368	63.157894736842	42.631578947368396	41.5789473684211	44.2105263157895
TRUE	50.52631578947237	45.7894736842105	44.736842105263216	43.1578947368421	50.52631578947237	45.263157894736	50.52631578947237	45.7894736842105	44.2105263157895
TTW	42.631578947368396	43.1578947368421	43.1578947368421	43.1578947368421	43.1578947368421	38.4210526315789	42.631578947368396	43.1578947368421	43.1578947368421
TU	49.473684210263	55.263157894736	45.263157894736	53.1578947368421	56.842105263157904	61.052631578947405	48.421052631578895	55.263157894736	45.263157894736
WHA	50.0	50.0	44.7368421052632	49.473684210263	51.052631578947405	57.368421052631604	50.0	50.0	44.7368421052632

Table 115 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	43.684210526315795	43.684210526315795	43.684210526315795	56.3157894736842	51.578947368421105	56.3157894736842	43.684210526315795	43.684210526315795	43.684210526315795
AOT	38.9473684210526	38.9473684210526	38.9473684210526	61.052631578947405	61.052631578947405	46.3157894736842	38.9473684210526	38.9473684210526	38.9473684210526
BA	0.0	0.0	0.0	0.0	0.0	0.0	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	41.57894736842105	41.57894736842105	41.57894736842105	55.2631578947368	44.736842105263156	55.26315789473684	41.57894736842105	41.57894736842105	41.57894736842105
BBL	44.736842105263156	44.736842105263156	44.736842105263156	44.2105263157895	44.736842105263156	55.78947368421053	44.736842105263156	44.736842105263156	44.736842105263156
BCP	38.42105263157895	38.42105263157895	38.42105263157895	61.57894736842105	61.57894736842105	48.421052631578895	38.42105263157895	38.42105263157895	38.42105263157895
BDMS	52.631578947368396	37.368421052631604	37.368421052631604	37.368421052631604	47.36842105263156	37.368421052631604	52.631578947368396	37.368421052631604	52.631578947368396
BEC	48.421052631578895	48.421052631578895	48.421052631578895	45.26315789473684	45.26315789473684	48.421052631578895	48.421052631578895	48.421052631578895	48.421052631578895
BH	49.47368421052632	49.47368421052632	49.47368421052632	50.52631578947368	43.157894736842105	49.473684210526315	49.473684210526315	49.473684210526315	49.473684210526315
BLA	45.78947368421053	45.78947368421053	45.78947368421053	56.3157894736842	54.21052631578947	45.78947368421053	45.78947368421053	45.78947368421053	45.78947368421053
BTS	40.52631578947368	40.52631578947368	40.52631578947368	40.52631578947368	59.473684210526315	56.842105263157904	40.52631578947368	40.52631578947368	40.52631578947368
CBG	32.0	32.0	32.0	68.0	68.0	68.0	32.0	32.0	32.0
CENTEL	41.57894736842105	41.57894736842105	41.57894736842105	58.421052631578995	58.421052631578995	40.0	41.57894736842105	41.57894736842105	41.57894736842105
CK	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526
CPALL	43.157894736842105	43.157894736842105	43.157894736842105	56.842105263157904	51.578947368421105	56.842105263157904	43.157894736842105	43.157894736842105	43.157894736842105
CPF	43.684210526315795	43.684210526315795	43.684210526315795	56.3157894736842	56.3157894736842	56.3157894736842	43.684210526315795	43.684210526315795	43.684210526315795
CPN	38.42105263157895	38.42105263157895	38.42105263157895	47.36842105263156	61.57894736842111	61.57894736842111	38.42105263157895	38.42105263157895	38.42105263157895
DELTA	45.26315789473684	45.26315789473684	45.26315789473684	54.736842105263194	54.736842105263194	54.736842105263194	45.26315789473684	45.26315789473684	45.26315789473684
DTAC	54.21052631578955	50.0	50.0	50.0	54.210526315789	50.0	54.21052631578955	50.0	54.21052631578955
EGCO	38.9473684210526	38.9473684210526	38.9473684210526	44.736842105263	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526	38.9473684210526
GLOW	41.57894736842111	41.57894736842111	41.57894736842111	47.8947368421053	41.57894736842111	47.8947368421053	41.57894736842111	41.57894736842111	41.57894736842111
GPSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HMPRO	36.84210526315789	36.84210526315789	36.84210526315789	63.157894736842105	63.157894736842105	48.421052631578895	36.84210526315789	36.84210526315789	36.84210526315789
INTUCH	42.105263157894704	42.105263157894704	42.105263157894704	45.78947368421052	57.894736842105296	57.894736842105296	42.105263157894704	42.105263157894704	42.105263157894704
IRPC	43.157894736842105	43.157894736842105	43.157894736842105	43.157894736842105	52.1052631578947	47.8947368421053	43.157894736842105	43.157894736842105	43.157894736842105
IVL	42.631578947368396	42.631578947368396	42.631578947368396	42.631578947368396	57.368421052631604	57.368421052631604	42.631578947368396	42.631578947368396	42.631578947368396
KBANK	45.26315789473684	45.26315789473684	45.26315789473684	45.26315789473684	50.0	45.26315789473684	45.26315789473684	45.26315789473684	45.26315789473684
KCE	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0
KTB	39.473684210526315	39.473684210526315	39.473684210526315	39.473684210526315	60.526315789473694	60.526315789473694	39.473684210526315	39.473684210526315	39.473684210526315
LH	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895	52.631578947368396	47.368421052631616	38.42105263157895	38.42105263157895	38.42105263157895
MINT	42.105263157894704	42.105263157894704	42.105263157894704	57.894736842105296	42.105263157894704	57.894736842105296	42.105263157894704	42.105263157894704	42.105263157894704
MTLS	0.0	0.0	0.0	0.0	0.0	0.0	44.89795918367535	44.89795918367535	44.89795918367535
PS	36.3157894736842	36.3157894736842	36.3157894736842	53.684210526315	53.684210526315	53.684210526315	36.3157894736842	36.3157894736842	36.3157894736842
PTT	56.3157894736842	58.9473684210526	58.9473684210526	58.9473684210526	56.3157894736842	58.9473684210526	56.3157894736842	58.9473684210526	56.3157894736842
PTTEP	54.21052631578955	54.21052631578955	54.21052631578955	52.631578947368396	47.368421052631616	47.368421052631616	54.21052631578955	54.21052631578955	54.21052631578955
PTTGC	46.842105263157904	46.842105263157904	46.842105263157904	53.157894736842105	48.947368421052616	48.947368421052616	46.842105263157904	46.842105263157904	46.842105263157904
ROBINS	37.894736842105296	37.894736842105296	37.894736842105296	48.421052631578895	62.105263157894704	62.105263157894704	37.894736842105296	37.894736842105296	37.894736842105296
SAWAD	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111	44.44444444444444	41.11111111111111	41.11111111111111	41.11111111111111	41.11111111111111
SCB	48.421052631578895	41.052631578947474	41.052631578947474	41.052631578947474	41.052631578947474	48.421052631578895	48.421052631578895	41.052631578947474	48.421052631578895
SCC	40.0	40.0	40.0	60.0	40.0	46.842105263157904	40.0	40.0	40.0
TASCO	44.736842105263156	46.3157894736842	46.3157894736842	53.684210526315	44.736842105263156	53.684210526315	44.736842105263156	46.3157894736842	44.736842105263156
TCAP	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895	38.42105263157895
TMB	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	56.3157894736842	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795
TOP	40.0	40.0	40.0	60.0	60.0	60.0	40.0	40.0	40.0
TPIPL	44.21052631578955	44.21052631578955	44.21052631578955	44.21052631578955	55.78947368421055	55.78947368421055	44.21052631578955	44.21052631578955	44.21052631578955
TRUE	45.26315789473684	45.26315789473684	45.26315789473684	53.684210526315	45.26315789473684	45.26315789473684	45.26315789473684	45.26315789473684	45.26315789473684
TTW	34.736842105263156	34.736842105263156	34.736842105263156	47.8947368421053	47.8947368421053	34.736842105263156	34.736842105263156	34.736842105263156	34.736842105263156
TU	40.0	40.0	40.0	52.631578947368396	52.631578947368396	60.0	40.0	40.0	40.0
WHA	44.736842105263156	44.736842105263156	44.736842105263156	42.631578947368396	57.368421052631604	57.368421052631604	44.736842105263156	44.736842105263156	44.736842105263156

PredictDirection17 : EMA Change

Table 116 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.21052631578955	43.684210526315795	43.684210526315795	43.1578947368421	44.21052631578955	44.21052631578955	44.736842105263156	43.684210526315795	43.684210526315795
AOT	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0
BA	44.44444444444444	57.4074074074074	57.4074074074074	48.148148148148	59.25925925925925	59.25925925925925	42.592592592592595	50.0	59.25925925925925
BANPU	41.57894736842111	41.57894736842111	38.947368421052616	41.57894736842111	41.57894736842111	41.052631578947474	41.57894736842111	41.57894736842111	38.947368421052616
BBL	45.78947368421055	44.736842105263156	44.736842105263156	46.31578947368421	45.26315789473684	45.78947368421055	46.31578947368421	44.736842105263156	44.210526315789474
BCP	59.473684210526315	58.421052631578995	56.31578947368421	55.26315789473684	57.368421052631604	57.368421052631604	57.894736842105296	57.894736842105296	56.31578947368421
BDMS	60.526315789473694	60.526315789473694	58.421052631578995	61.052631578947405	61.052631578947405	57.368421052631604	60.0	60.526315789473694	58.421052631578995
BEC	45.26315789473684	44.736842105263156	47.368421052631616	44.736842105263156	44.736842105263156	47.368421052631616	45.78947368421055	44.736842105263156	47.368421052631616
BH	52.631578947368396	50.526315789473705	52.105263157894704	52.105263157894704	50.526315789473705	50.0	51.052631578947405	50.526315789473705	51.052631578947405
BLA	48.947368421052616	49.473684210526315	48.421052631578895	47.368421052631616	49.473684210526315	47.894736842105263	47.894736842105263	49.473684210526315	49.473684210526315
BTS	45.26315789473684	48.421052631578895	45.78947368421055	44.736842105263156	46.842105263157904	46.842105263157904	45.26315789473684	46.842105263157904	45.78947368421055
CBG	70.0	70.0	68.0	68.0	70.0	68.0	70.0	68.0	68.0
CENTEL	58.421052631578995	58.421052631578995	57.368421052631604	58.421052631578995	58.947368421052616	58.421052631578995	58.421052631578995	58.421052631578995	57.368421052631604
CK	57.894736842105296	61.57894736842111	56.842105263157904	58.947368421052616	62.105263157894704	58.947368421052616	58.421052631578995	61.052631578947405	56.31578947368421
CPALL	55.26315789473684	54.21052631578955	55.78947368421055	55.263157894736805	53.15789473684211	51.578947368421105	55.26315789473684	54.21052631578955	55.78947368421055
CPF	48.947368421052616	50.0	47.894736842105263	50.0	48.421052631578895	49.473684210526315	48.947368421052616	48.421052631578895	47.894736842105263
CPN	55.78947368421055	55.78947368421055	57.894736842105296	55.263157894736805	56.31578947368421	57.368421052631604	55.26315789473684	55.26315789473684	57.368421052631604
DELTA	50.0	51.052631578947405	52.105263157894704	52.631578947368396	51.578947368421105	53.684210526315	51.578947368421105	51.578947368421105	51.578947368421105
DTAC	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
EGCO	57.368421052631604	57.894736842105296	55.78947368421055	58.421052631578995	58.947368421052616	55.263157894736805	57.368421052631604	58.421052631578995	55.78947368421055
GLOW	46.31578947368421	45.78947368421055	50.0	47.368421052631616	47.894736842105263	49.473684210526315	45.78947368421055	45.78947368421055	50.0
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	55.78947368421055	52.105263157894704	55.263157894736805	53.15789473684211	52.631578947368396	55.78947368421055	53.684210526315	52.631578947368396	55.263157894736805
INTUCH	42.105263157894704	43.684210526315795	41.052631578947474	42.105263157894704	42.631578947368396	41.578947368421104	42.105263157894704	43.684210526315795	41.052631578947474
IRPC	54.736842105263194	54.736842105263194	56.842105263157904	53.684210526315	54.736842105263194	56.315789473684205	54.736842105263194	54.736842105263194	57.368421052631604
IVL	55.263157894736805	54.210526315789474	55.263157894736805	54.736842105263194	53.684210526315	52.105263157894704	54.736842105263194	54.210526315789474	54.736842105263194
KBANK	45.263157894736805	43.684210526315795	45.78947368421055	45.263157894736805	44.210526315789474	45.263157894736805	45.78947368421055	44.210526315789474	45.78947368421055
KCE	60.0	60.0	60.0	60.0	59.473684210526315	60.0	59.473684210526315	60.0	60.0
KTB	41.57894736842111	41.052631578947474	42.631578947368396	41.57894736842111	41.052631578947474	41.052631578947474	41.57894736842111	41.57894736842111	42.631578947368396
LH	51.052631578947405	40.526315789473705	51.578947368421105	40.0	38.947368421052616	44.736842105263156	47.368421052631616	41.052631578947474	51.052631578947405
MINT	58.421052631578995	58.947368421052616	58.947368421052616	58.421052631578995	58.947368421052616	59.473684210526315	58.421052631578995	58.947368421052616	58.947368421052616
MTLS	55.102040816326505	55.102040816326505	53.06122448979591	53.06122448979591	53.06122448979591	53.06122448979591	55.102040816326505	53.06122448979591	53.06122448979591
PS	46.842105263157904	48.421052631578895	67.36842105263156	60.0	47.894736842105263	66.31578947368419	50.526315789473705	47.368421052631616	67.36842105263156
PTT	42.631578947368396	43.157894736842105	40.0	41.578947368421104	42.631578947368396	41.578947368421104	42.105263157894704	42.631578947368396	40.0
PTTEP	47.36842105263156	47.36842105263156	46.842105263157904	47.36842105263156	47.36842105263156	46.842105263157904	47.36842105263156	47.36842105263156	46.842105263157904
PTTGC	47.894736842105263	45.789473684210505	52.631578947368396	44.736842105263232	45.263157894736805	52.631578947368396	44.736842105263232	46.31578947368421	52.631578947368396
ROBINS	48.947368421052616	48.421052631578895	52.631578947368396	50.526315789473705	48.421052631578895	47.36842105263156	49.473684210526315	48.421052631578895	52.631578947368396
SAWAD	48.888888888888889	52.222222222222222	51.11111111111109	54.44444444444439	53.333333333333	52.222222222222222	52.222222222222222	52.222222222222222	51.11111111111109
SCB	40.526315789473705	41.052631578947474	42.105263157894704	40.526315789473705	41.052631578947474	42.631578947368396	40.526315789473705	41.578947368421104	42.105263157894704
SCC	45.789473684210505	46.842105263157904	48.421052631578895	44.736842105263232	48.421052631578895	51.578947368421105	45.263157894736805	50.0	48.421052631578895
TASCO	53.684210526315	53.684210526315	53.684210526315	53.684210526315	53.684210526315	53.684210526315	53.684210526315	53.684210526315	53.684210526315
TCAP	54.736842105263194	52.631578947368396	54.210526315789474	54.210526315789474	52.631578947368396	54.210526315789474	55.263157894736805	52.105263157894704	54.210526315789474
TMB	57.368421052631604	54.210526315789474	55.263157894736805	56.842105263157904	52.631578947368396	56.315789473684205	58.421052631578995	52.631578947368396	55.263157894736805
TOP	60.0	60.526315789473694	59.473684210526315	60.0	60.0	60.0	60.0	60.526315789473694	59.473684210526315
TPIPL	49.473684210526315	55.263157894736805	59.473684210526315	49.473684210526315	45.263157894736805	57.368421052631604	49.473684210526315	54.210526315789474	59.473684210526315
TRUE	55.78947368421055	51.578947368421105	56.315789473684205	47.894736842105263	51.052631578947405	54.736842105263194	54.736842105263194	54.736842105263194	56.315789473684205
TTW	54.21052631578955	55.263157894736805	54.736842105263194	44.210526315789474	60.0	56.315789473684205	51.052631578947405	59.473684210526315	54.736842105263194
TU	45.789473684210505	44.736842105263232	49.473684210526315	45.789473684210505	45.789473684210505	51.052631578947405	47.368421052631616	46.315789473684205	49.473684210526315
WHA	52.105263157894704	59.473684210526315	56.842105263157904	54.210526315789474	59.473684210526315	55.78947368421055	51.578947368421105	59.473684210526315	56.315789473684205

Table 117 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	48.421052631578895	46.842105263157904	45.2631578947368	48.421052631578895	43.684210526315795	45.26315789473684	47.89473684210553	46.31578947368421	45.78947368421053
AOT	50.52631578947368	44.736842105263216	47.89473684210553	43.157894736842105	41.05263157894737	41.05263157894737	41.05263157894737	51.578947368421105	44.736842105263216
BA	40.74074074074074	38.88888888888889	38.88888888888889	38.88888888888889	35.185185185185205	42.592592592592595	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	42.631578947368396	47.36842105263216	42.631578947368396	51.052631578947405	49.473684210526316	46.842105263157904	42.631578947368396	46.842105263157904	42.631578947368396
BBL	43.15789473684211	42.105263157894704	37.894736842105296	45.2631578947368	44.736842105263216	43.15789473684211	45.26315789473684	42.105263157894704	38.94736842105263
BCP	45.78947368421053	46.31578947368421	42.631578947368396	44.21052631578947	43.15789473684211	45.26315789473684	46.31578947368421	49.473684210526316	42.631578947368396
BDMS	46.31578947368421	45.26315789473684	48.94736842105263	41.05263157894737	39.473684210526316	41.57894736842105	44.21052631578947	45.78947368421053	50.0
BEC	50.0	48.421052631578895	54.21052631578947	50.0	48.94736842105263	47.36842105263216	50.0	48.94736842105263	54.21052631578947
BH	52.631578947368396	51.052631578947405	45.2631578947368	50.52631578947368	49.473684210526316	46.31578947368421	53.68421052631579	45.78947368421053	45.26315789473684
BLA	50.52631578947368	47.89473684210553	53.157894736842105	52.631578947368396	53.684210526315795	51.052631578947405	50.0	47.89473684210553	51.052631578947405
BTS	39.473684210526316	40.52631578947368	40.0	42.105263157894704	41.05263157894737	42.105263157894704	40.0	40.52631578947368	41.05263157894737
CBG	46.0	56.000000000000001	57.999999999999999	46.0	57.999999999999999	57.999999999999999	42.0	57.999999999999999	57.999999999999999
CENTEL	48.94736842105263	49.473684210526316	46.842105263157904	43.684210526315795	42.631578947368396	41.57894736842105	50.0	47.36842105263216	46.31578947368421
CK	43.684210526315795	47.36842105263216	41.05263157894737	43.684210526315795	44.736842105263216	44.21052631578947	43.684210526315795	47.36842105263216	41.05263157894737
CPALL	52.10526315789474	49.473684210526316	49.473684210526316	49.473684210526316	47.894736842105263	43.684210526315795	52.10526315789474	48.421052631578895	49.473684210526316
CPF	55.78947368421053	54.736842105263216	52.631578947368396	54.736842105263216	54.736842105263216	50.52631578947368	55.78947368421053	55.78947368421053	52.10526315789474
CPN	50.0	51.052631578947405	51.052631578947405	39.473684210526316	41.57894736842105	41.05263157894737	52.10526315789474	51.052631578947405	51.052631578947405
DELTA	51.578947368421105	43.684210526315795	43.684210526315795	45.2631578947368	45.26315789473684	45.26315789473684	52.10526315789474	45.26315789473684	43.684210526315795
DTAC	49.473684210526316	51.052631578947405	51.052631578947405	54.736842105263216	53.157894736842105	55.78947368421053	49.473684210526316	51.052631578947405	50.52631578947368
EGCO	46.842105263157904	44.21052631578947	43.684210526315795	46.31578947368421	48.94736842105263	42.105263157894704	46.31578947368421	44.736842105263216	42.631578947368396
GLOW	47.36842105263216	46.31578947368421	40.52631578947368	42.631578947368396	41.05263157894737	43.157894736842105	47.89473684210553	45.78947368421053	40.52631578947368
GPSC	25.0	43.75	43.75	43.75	43.75	43.75	31.25	37.5	43.75
HMPRO	58.94736842105263	55.78947368421053	55.26315789473684	56.31578947368421	55.78947368421053	50.0	57.368421052631604	55.78947368421053	55.26315789473684
INTUCH	43.15789473684211	42.105263157894704	42.105263157894704	43.157894736842105	43.157894736842105	41.05263157894737	43.157894736842105	41.57894736842105	42.105263157894704
IRPC	55.26315789473684	56.842105263157904	55.26315789473684	51.578947368421105	53.157894736842105	53.684210526315795	54.736842105263216	56.31578947368421	55.26315789473684
IVL	47.89473684210553	50.0	53.157894736842105	48.94736842105263	49.473684210526316	51.052631578947405	46.31578947368421	49.473684210526316	53.157894736842105
KBANK	51.052631578947405	52.10526315789474	44.736842105263216	44.21052631578947	43.684210526315795	44.736842105263216	52.10526315789474	51.578947368421105	44.21052631578947
KCE	53.684210526315795	51.578947368421105	57.894736842105296	48.421052631578895	45.78947368421053	52.10526315789474	53.157894736842105	51.578947368421105	57.894736842105296
KTB	51.578947368421105	49.473684210526316	39.473684210526316	44.21052631578947	49.473684210526316	43.684210526315795	52.10526315789474	48.421052631578895	40.0
LH	48.421052631578895	44.736842105263216	54.21052631578947	43.684210526315795	44.736842105263216	47.89473684210553	50.0	45.78947368421053	54.736842105263216
MINT	52.631578947368396	54.736842105263216	46.842105263157904	45.78947368421053	44.736842105263216	41.57894736842105	52.631578947368396	54.21052631578947	46.842105263157904
MTLS	48.9795918367347	46.93877551020408	53.06122448979592	46.93877551020408	48.9795918367347	48.9795918367347	48.9795918367347	46.93877551020408	53.06122448979592
PS	47.89473684210553	44.21052631578947	48.947368421052636	37.894736842105296	37.368421052631604	40.52631578947368	48.421052631578895	44.21052631578947	48.947368421052636
PTT	53.157894736842105	51.578947368421105	55.78947368421053	46.842105263157904	43.157894736842105	41.05263157894737	52.631578947368396	51.578947368421105	54.736842105263219
PTTEP	53.684210526315795	52.631578947368396	52.10526315789474	51.578947368421105	53.684210526315795	50.52631578947368	52.631578947368396	52.631578947368396	52.10526315789474
PTTGC	50.0	50.0	56.842105263157904	53.157894736842105	52.10526315789474	56.31578947368421	49.473684210526316	50.0	56.31578947368421
ROBINS	53.157894736842105	57.368421052631604	53.684210526315795	45.26315789473684	52.10526315789474	50.52631578947368	52.631578947368396	57.368421052631604	53.684210526315795
SAWAD	43.333333333333333	47.77777777777778	46.666666666666667	40.0	41.111111111111111	41.111111111111111	46.666666666666667	47.77777777777778	47.77777777777778
SCB	53.684210526315795	51.578947368421105	50.0	48.421052631578895	50.526315789473684	45.26315789473684	53.684210526315795	51.578947368421105	50.0
SCC	39.473684210526316	39.473684210526316	40.52631578947368	40.52631578947368	40.0	41.57894736842105	42.105263157894704	40.0	40.52631578947368
TASCO	45.78947368421053	47.36842105263216	49.473684210526316	47.368421052631616	46.31578947368421	46.842105263157904	47.89473684210553	47.36842105263216	49.473684210526316
TCAP	47.89473684210553	46.31578947368421	47.89473684210553	53.157894736842105	54.21052631578947	48.421052631578895	47.89473684210553	46.842105263157904	47.89473684210553
TMB	47.89473684210553	50.0	42.105263157894704	47.89473684210553	51.052631578947405	46.31578947368421	47.89473684210553	49.473684210526316	42.105263157894704
TOP	49.473684210526316	51.052631578947405	47.89473684210553	52.631578947368396	54.736842105263219	52.10526315789474	50.0	51.052631578947405	48.421052631578895
TPIPL	42.105263157894704	43.684210526315795	40.52631578947368	47.89473684210553	48.421052631578895	46.31578947368421	43.157894736842105	42.631578947368396	40.52631578947368
TRUE	42.631578947368396	45.78947368421053	56.31578947368421	55.78947368421053	52.10526315789474	55.26315789473684	43.157894736842105	46.842105263157904	56.31578947368421
TTW	48.421052631578895	38.42105263157895	41.05263157894737	42.631578947368396	38.42105263157895	41.57894736842105	47.36842105263216	37.894736842105296	41.05263157894737
TU	47.89473684210553	47.89473684210553	47.89473684210553	57.368421052631604	53.684210526315795	36.842105263157895	46.31578947368421	47.89473684210553	47.89473684210553
WHA	46.31578947368421	46.31578947368421	43.157894736842105	45.26315789473684	44.736842105263216	44.736842105263216	47.36842105263216	46.31578947368421	42.631578947368396

Table 118 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	44.21052631578947	43.684210526315795	47.36842105263216	46.842105263157904	43.15789473684211	43.15789473684211	43.684210526315795	44.21052631578947	46.31578947368421
AOT	53.157894736842105	47.36842105263216	50.0	45.26315789473684	45.78947368421053	40.0	50.52631578947368	47.36842105263216	50.0
BA	38.88888888888889	38.88888888888889	38.88888888888889	35.185185185185205	35.185185185185205	38.88888888888889	37.037037037037	38.88888888888889	38.88888888888889
BANPU	43.684210526315795	47.89473684210553	44.736842105263232	48.94736842105263	49.473684210526316	47.89473684210553	42.631578947368396	47.36842105263216	44.736842105263232
BBL	42.105263157894704	41.57894736842111	38.42105263157895	45.78947368421053	44.21052631578947	42.631578947368396	43.157894736842105	42.105263157894704	38.42105263157895
BCP	46.31578947368421	48.421052631578895	42.631578947368396	45.78947368421053	42.631578947368396	45.26315789473684	46.842105263157904	48.421052631578895	42.631578947368396
BDMS	45.78947368421053	42.631578947368396	49.473684210526316	41.05263157894737	38.42105263157895	41.57894736842111	45.26315789473684	45.26315789473684	50.0
BEC	50.0	47.89473684210553	54.736842105263219	48.94736842105263	45.78947368421053	47.36842105263216	51.052631578947405	47.36842105263216	54.736842105263219
BH	51.578947368421105	44.21052631578947	50.0	48.421052631578895	47.368421052631616	49.473684210526316	51.578947368421105	47.36842105263216	48.94736842105263
BLA	51.052631578947405	48.421052631578895	53.157894736842105	54.21052631578947	53.684210526315795	51.052631578947405	49.473684210526316	48.421052631578895	51.052631578947405
BTS	42.631578947368396	43.157894736842105	40.0	41.05263157894737	41.05263157894737	41.57894736842111	41.052631578947405	41.57894736842111	40.52631578947368
CBG	52.0	60.0	57.999999999999999	64.0	50.0	57.999999999999999	52.0	70.0	57.999999999999999
CENTEL	48.94736842105263	50.0	45.78947368421053	44.736842105263232	44.21052631578947	42.105263157894704	50.52631578947368	49.473684210526316	46.31578947368421
CK	47.36842105263216	47.89473684210553	40.52631578947368	52.631578947368396	51.052631578947405	45.78947368421053	44.21052631578947	47.36842105263216	40.52631578947368
CPALL	52.10526315789474	48.94736842105263	49.473684210526316	48.421052631578895	47.894736842105263	43.684210526315795	50.0	48.421052631578895	49.473684210526316
CPF	52.631578947368396	53.157894736842105	48.94736842105263	46.842105263157904	53.684210526315795	48.94736842105263	52.631578947368396	53.157894736842105	50.0
CPN	53.684210526315795	51.052631578947405	51.052631578947405	39.473684210526316	41.05263157894737	41.05263157894737	52.631578947368396	51.052631578947405	51.052631578947405
DELTA	52.10526315789474	46.31578947368421	43.684210526315795	45.26315789473684	45.26315789473684	45.26315789473684	51.578947368421105	44.21052631578947	43.684210526315795
DTAC	48.421052631578895	52.631578947368396	49.473684210526316	51.578947368421105	48.421052631578895	51.052631578947405	51.578947368421105	52.631578947368396	48.94736842105263
EGCO	45.78947368421053	42.631578947368396	43.684210526315795	45.78947368421053	46.31578947368421	42.631578947368396	46.842105263157904	43.684210526315795	42.631578947368396
GLOW	47.89473684210553	45.78947368421053	40.52631578947368	42.105263157894704	41.05263157894737	43.157894736842105	47.89473684210553	46.31578947368421	40.52631578947368
GPSC	37.5	37.5	43.75	43.75	43.75	43.75	31.25	31.25	43.75
HMPRO	59.473684210526316	55.78947368421053	55.26315789473684	57.894736842105296	58.421052631578995	50.52631578947368	55.26315789473684	55.26315789473684	55.26315789473684
INTUCH	43.15789473684211	42.105263157894704	42.105263157894704	43.157894736842105	43.157894736842105	41.05263157894737	43.684210526315795	42.105263157894704	42.105263157894704
IRPC	54.21052631578947	56.31578947368421	54.736842105263219	54.21052631578947	53.684210526315795	52.631578947368396	55.78947368421053	55.78947368421053	54.736842105263219
IVL	47.89473684210553	48.421052631578895	50.0	47.89473684210553	47.368421052631616	51.052631578947405	48.94736842105263	50.52631578947368	50.0
KBANK	48.421052631578895	46.31578947368421	46.31578947368421	49.473684210526316	47.894736842105263	51.052631578947405	48.94736842105263	46.31578947368421	46.31578947368421
KCE	54.21052631578947	52.10526315789474	55.78947368421053	46.31578947368421	43.684210526315795	52.631578947368396	54.736842105263219	52.10526315789474	55.78947368421053
KTB	53.684210526315795	47.36842105263216	39.473684210526316	44.21052631578947	51.052631578947405	53.684210526315795	52.631578947368396	46.842105263157904	40.0
LH	47.36842105263216	47.36842105263216	54.736842105263219	52.631578947368396	55.26315789473684	51.052631578947405	45.78947368421053	47.36842105263216	55.26315789473684
MINT	56.31578947368421	54.736842105263219	39.473684210526316	54.736842105263219	53.157894736842105	42.631578947368396	56.31578947368421	53.684210526315795	39.473684210526316
MTLS	48.9795918367347	48.9795918367347	53.06122448979592	46.93877551020408	48.9795918367347	48.9795918367347	48.9795918367347	46.93877551020408	53.06122448979592
PS	46.31578947368421	42.631578947368396	46.31578947368421	49.473684210526316	49.473684210526316	37.368421052631604	46.842105263157904	42.105263157894704	46.31578947368421
PTT	53.157894736842105	52.10526315789474	55.78947368421053	50.0	41.578947368421105	41.05263157894737	53.157894736842105	51.578947368421105	54.736842105263219
PTTEP	51.052631578947405	51.578947368421105	52.631578947368396	46.31578947368421	46.842105263157904	45.78947368421053	50.52631578947368	51.578947368421105	52.631578947368396
PTTGC	49.473684210526316	49.473684210526316	56.842105263157904	53.157894736842105	51.578947368421105	56.31578947368421	50.0	50.0	56.31578947368421
ROBINS	52.631578947368396	57.368421052631604	53.684210526315795	44.21052631578947	51.578947368421105	50.52631578947368	54.21052631578947	57.368421052631604	53.684210526315795
SAWAD	42.222222222222	48.88888888888889	46.666666666666667	41.111111111111111	41.111111111111111	41.111111111111111	45.555555555556	47.77777777777778	47.77777777777778
SCB	54.736842105263219	51.052631578947405	50.0	49.473684210526316	50.0	45.26315789473684	53.157894736842105	51.052631578947405	50.0
SCC	40.52631578947368	40.52631578947368	41.05263157894737	43.157894736842105	37.894736842105296	42.631578947368396	38.94736842105263	40.0	41.05263157894737
TASCO	48.94736842105263	50.0	52.631578947368396	48.94736842105263	46.31578947368421	46.842105263157904	50.52631578947368	50.0	52.631578947368396
TCAP	49.473684210526316	46.31578947368421	47.89473684210553	52.10526315789474	51.578947368421105	48.421052631578895	48.421052631578895	45.26315789473684	47.89473684210553
TMB	47.89473684210553	48.421052631578895	45.26315789473684	48.421052631578895	51.10526315789474	48.421052631578895	47.89473684210553	48.94736842105263	44.736842105263232
TOP	47.89473684210553	50.52631578947368	47.89473684210553	54.736842105263219	54.736842105263219	52.10526315789474	49.473684210526316	50.0	48.421052631578895
TPIPL	40.0	43.157894736842105	42.105263157894704	46.842105263157904	45.78947368421053	45.78947368421053	42.631578947368396	43.157894736842105	42.105263157894704
TRUE	43.684210526315795	44.736842105263232	53.684210526315795	41.57894736842111	40.0	45.78947368421053	44.21052631578947	44.21052631578947	53.684210526315795
TTW	41.57894736842111	38.42105263157895	39.473684210526316	37.894736842105296	40.0	34.736842105263232	50.0	38.42105263157895	40.0
TU	49.473684210526316	47.36842105263216	47.89473684210553	58.94736842105263	58.94736842105263	36.842105263157895	50.0	46.31578947368421	47.89473684210553
WHA	46.31578947368421	45.78947368421053	42.105263157894704	46.31578947368421	44.21052631578947	45.26315789473684	46.842105263157904	45.78947368421053	42.631578947368396

Table 119 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	41.0526315789474	41.57894736842211	43.1578947368421	46.84210526315790	50.0	43.68421052631579	42.1052631578947	41.57894736842211	43.1578947368421
AOT	42.1052631578947	46.84210526315790	38.4210526315789	41.5789473684211	46.31578947368421	48.42105263157895	43.68421052631579	45.78947368421053	39.47368421052632
BA	38.88888888888889	38.88888888888889	38.88888888888889	53.703703703703695	42.59259259259259	42.59259259259259	38.88888888888889	38.88888888888889	38.88888888888889
BANPU	42.1052631578947	45.26315789473684	41.5789473684211	54.73684210526319	45.78947368421053	52.1052631578947	43.1578947368421	43.1578947368421	41.5789473684211
BBL	41.5789473684211	41.0526315789474	44.7368421052632	51.57894736842105	45.26315789473684	53.1578947368421	40.0	41.5789473684211	44.7368421052632
BCP	42.1052631578947	42.1052631578947	37.368421052631604	48.94736842105263	48.42105263157895	44.2105263157895	41.5789473684211	42.1052631578947	37.368421052631604
BDMS	38.9473684210526	38.9473684210526	36.8421052631579	47.8947368421053	42.631578947368396	51.052631578947405	38.9473684210526	39.4736842105263	36.8421052631579
BEC	45.26315789473684	44.7368421052632	42.631578947368396	50.5263157894737	52.1052631578947	48.421052631578895	45.26315789473684	45.26315789473684	42.631578947368396
BH	50.0	48.421052631578895	49.4736842105263	46.31578947368421	48.94736842105263	50.526315789473	50.0	49.4736842105263	49.4736842105263
BLA	42.631578947368396	41.57894736842211	45.78947368421053	43.1578947368421	41.57894736842211	52.1052631578947	42.631578947368396	42.631578947368396	45.78947368421053
BTS	40.52631578947737	39.47368421052632	39.47368421052632	38.4210526315789	39.473684210526	41.57894736842211	40.0	39.47368421052632	39.47368421052632
CBG	34.0	36.0	42.0	38.0	50.0	48.0	34.0	42.0	42.0
CENTEL	41.0526315789474	40.52631578947737	39.47368421052632	59.473684210526	50.52631578947737	44.2105263157895	41.0526315789474	38.4210526315789	39.47368421052632
CK	38.4210526315789	38.94736842105263	38.94736842105263	45.78947368421053	45.78947368421053	53.1578947368421	38.4210526315789	38.94736842105263	38.94736842105263
CPALL	43.1578947368421	43.1578947368421	43.1578947368421	45.26315789473684	47.36842105263158	52.1052631578947	43.1578947368421	42.631578947368396	43.1578947368421
CPF	47.89473684210526	50.52631578947737	43.684210526315795	57.368421052631604	55.26315789473684	54.2105263157895	48.421052631578895	49.473684210526	46.842105263157904
CPN	50.52631578947737	49.47368421052632	45.26315789473684	51.052631578947405	43.684210526315795	52.1052631578947	51.052631578947405	50.52631578947737	44.7368421052632
DELTA	45.26315789473684	45.26315789473684	45.26315789473684	48.421052631578895	45.78947368421053	52.1052631578947	45.26315789473684	45.26315789473684	45.26315789473684
DTAC	48.9473684210526	53.1578947368421	49.4736842105263	48.421052631578895	48.94736842105263	48.57894736842105	47.36842105263158	52.631578947368396	50.0
EGCO	39.473684210526	41.0526315789474	38.4210526315789	47.89473684210553	44.736842105263	52.631578947368	40.0	41.0526315789474	38.4210526315789
GLOW	42.631578947368396	42.105263157894704	42.105263157894704	44.2105263157895	42.105263157894704	51.578947368421105	42.631578947368396	40.0	42.631578947368396
GPSC	37.5	43.75	43.75	37.5	37.5	37.5	43.75	43.75	43.75
HMPRO	43.1578947368421	46.3157894736842	36.8421052631579	50.5263157894737	50.5263157894737	48.421052631578895	43.684210526315795	46.3157894736842	36.8421052631579
INTUCH	42.105263157894704	42.105263157894704	42.105263157894704	46.84210526315790	44.2105263157895	43.684210526315795	42.105263157894704	42.105263157894704	42.105263157894704
IRPC	50.0	40.0	42.105263157894704	53.6842105263158	51.578947368421105	45.26315789473684	43.684210526315795	41.0526315789474	42.105263157894704
IVL	47.89473684210553	43.684210526315795	41.5789473684211	52.1052631578947	47.89473684210553	45.26315789473684	46.3157894736842	43.15789473684211	41.5789473684211
KBANK	46.84210526315790	45.78947368421053	45.26315789473684	51.052631578947405	53.1578947368421	48.421052631578895	46.84210526315790	47.36842105263158	45.26315789473684
KCE	50.52631578947737	43.1578947368421	40.0	51.578947368421105	50.52631578947737	46.842105263157904	52.1052631578947	45.26315789473684	40.0
KTB	39.473684210526	39.473684210526	39.473684210526	51.052631578947405	48.94736842105263	55.26315789473684	39.473684210526	39.473684210526	39.473684210526
LH	38.9473684210526	36.31578947368421	38.4210526315789	46.3157894736842	47.36842105263158	56.31578947368421	38.4210526315789	35.26315789473684	38.4210526315789
MINT	40.52631578947737	40.52631578947737	40.52631578947737	52.52631578947737	48.94736842105263	48.94736842105263	41.578947368421	41.052631578947737	40.52631578947737
MTLS	44.89795918367535	48.97959183673347	48.97959183673347	46.938775102041	44.89795918367535	46.938775102041	46.938775102041	51.020408163265294	48.97959183673347
PS	36.84210526315789	36.84210526315789	36.31578947368421	47.36842105263158	50.0	45.78947368421053	36.84210526315789	36.31578947368421	36.31578947368421
PTT	44.7368421052632	44.2105263157895	41.05263157894737	48.9473684210526	52.1052631578947	53.15789473684211	46.31578947368421	42.105263157894704	41.05263157894737
PTTEP	55.263157894737	55.78947368421053	54.21052631578947	57.368421052631604	51.578947368421105	48.421052631578895	56.31578947368421	55.78947368421053	54.21052631578947
PTTGC	51.578947368421105	51.052631578947405	48.94736842105263	51.578947368421105	54.2105263157895	55.78947368421053	52.1052631578947	51.052631578947405	48.94736842105263
ROBINS	40.52631578947737	46.31578947368421	46.31578947368421	45.78947368421053	52.631578947368396	57.368421052631604	41.052631578947	50.0	46.31578947368421
SAWAD	46.66666666666667	45.55555555555556	42.22222222222222	52.22222222222222	48.88888888888889	52.22222222222222	45.55555555555556	47.77777777777778	44.44444444444444
SCB	46.31578947368421	46.31578947368421	41.57894736842211	49.473684210526	46.31578947368421	51.578947368421105	46.842105263157904	45.78947368421053	41.57894736842211
SCC	39.47368421052632	38.9473684210526	40.52631578947737	47.8947368421053	41.57894736842211	44.2105263157895	39.473684210526	38.4210526315789	40.52631578947737
TASCO	50.0	48.94736842105263	46.31578947368421	49.473684210526	48.421052631578895	51.578947368421105	49.473684210526	50.0	46.31578947368421
TCAP	39.47368421052632	37.368421052631604	38.4210526315789	49.473684210526	45.78947368421053	49.473684210526	36.842105263157904	37.368421052631604	38.4210526315789
TMB	43.684210526315795	43.684210526315795	43.684210526315795	41.052631578974	49.473684210526	57.368421052631604	43.15789473684211	43.684210526315795	43.684210526315795
TOP	41.5789473684211	41.0526315789474	43.1578947368421	53.1578947368421	49.473684210526	48.421052631578895	41.052631578974	42.631578947368396	43.1578947368421
TPIPL	44.7368421052632	45.26315789473684	44.2105263157895	42.631578947368396	42.105263157894704	47.36842105263158	43.684210526315795	44.7368421052632	44.2105263157895
TRUE	43.1578947368421	42.105263157894704	46.3157894736842	48.421052631578895	45.26315789473684	46.842105263157904	44.7368421052632	41.5789473684211	46.842105263157904
TTW	35.26315789473684	34.7368421052632	34.7368421052632	43.1578947368421	36.842105263579	36.842105263179	34.7368421052632	34.7368421052632	34.7368421052632
TU	50.0	46.842105263157904	40.0	51.052631578947405	54.2105263157895	50.0	47.36842105263216	46.3157894736842	40.0
WHA	51.052631578947405	47.36842105263216	44.7368421052632	46.842105263157904	45.26315789473684	54.2105263157895	50.526315789473	47.89473684210553	44.7368421052632

PredictDirection18 : EMA Change, Volume, and Buy-Sell Volume

Table 120 MAPE results from closing price.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	42.631578947368396	42.631578947368396	42.631578947368396	43.684210526315795	42.631578947368396	43.15789473684211	42.631578947368396	42.631578947368396	42.631578947368396
AOT	60.0	60.526315789473694	61.052631578947405	60.0	60.526315789473694	60.0	60.0	60.526315789473694	61.052631578947405
BA	53.703703703703695	62.962962962963	53.703703703703695	64.81481481481481	59.25925925925593	51.851851851851805	55.555555555556	61.111111111111111	55.555555555556
BANPU	38.4210526315789	40.0	40.526315789473737	40.0	41.052631578974	38.94736842105263	38.4210526315789	40.0	40.0
BBL	42.1052631578947	43.1578947368421	45.26315789473684	43.684210526315795	43.684210526315795	40.526315789473737	41.578947368421	43.1578947368421	45.26315789473684
BCP	56.31578947368421	59.473684210526	54.21052631578947	58.94736842105263	62.631578947368396	56.842105263157904	56.31578947368421	57.894736842105296	54.21052631578947
BDMS	58.9473684210526	61.578947368421	56.842105263157904	59.473684210526	61.578947368421	55.78947368421053	58.421052631578995	61.578947368421	56.842105263157904
BEC	44.210526315789	45.78947368421053	41.57894736842211	44.2105263157895	45.263157894737368	44.210526315789	45.263157894737368	45.78947368421053	41.57894736842211
BH	50.526315789473737	50.526315789473737	45.78947368421053	50.526315789473737	50.526315789473737	46.315789473684211	50.526315789473737	50.526315789473737	45.26315789473684
BLA	46.31578947368421	48.947368421052632	46.31578947368421	48.947368421052632	50.0	46.842105263157904	46.842105263157904	48.947368421052632	46.31578947368421
BTS	49.473684210526	51.052631578947405	46.31578947368421	48.421052631578895	51.578947368421105	42.631578947368396	49.473684210526	51.578947368421105	46.31578947368421
CBG	70.0	70.0	68.0	70.0	70.0	70.0	70.0	68.0	68.0
CENTEL	58.421052631578995	58.421052631578995	57.368421052631604	57.368421052631604	57.894736842105296	58.94736842105263	57.894736842105296	57.894736842105296	57.368421052631604
CK	58.421052631578995	59.473684210526	61.578947368421	61.578947368421	59.473684210526	60.0	57.894736842105296	59.473684210526	61.052631578947405
CPALL	54.73684210526319	54.73684210526319	52.105263157894	54.2105263157895	55.789473684210	53.684210526315	54.2105263157895	54.73684210526319	52.1052631578947
CPF	48.947368421052	45.263157894736	51.578947368421105	48.421052631578895	45.263157894736	52.1052631578947	48.947368421052	45.789473684210	51.052631578947405
CPN	57.894736842105296	58.947368421052	57.894736842105296	56.842105263157904	57.894736842105296	56.842105263157904	57.894736842105296	58.947368421052	57.894736842105296
DELTA	53.1578947368421	55.263157894736	46.842105263157904	54.2105263157895	56.315789473684	47.368421052631	53.1578947368421	57.368421052631604	46.31578947368421
DTAC	50.526315789473737	50.526315789473737	50.526315789473737	50.526315789473737	50.526315789473737	51.052631578974405	50.526315789473737	50.526315789473737	50.526315789473737
EGCO	57.894736842105296	57.894736842105296	57.368421052631604	58.947368421052	58.947368421052	57.894736842105296	57.894736842105296	58.421052631578995	57.368421052631604
GLOW	51.052631578947405	52.631578947368396	52.631578947368396	50.526315789473737	50.0	47.89473684210553	51.052631578947405	53.157894736841	52.631578947368396
GPSC	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
HMPRO	50.526315789473737	50.526315789473737	54.73684210526319	49.473684210526	48.421052631578895	53.684210526315	51.578947368421105	50.0	54.73684210526319
INTUCH	40.0	40.526315789473737	41.052631578974	40.526315789473737	40.526315789473737	41.052631578974	40.0	40.526315789473737	41.052631578974
IRPC	56.31578947368421	57.368421052631604	53.684210526315	56.31578947368421	56.315789473684	55.789473684210	56.315789473684	57.368421052631604	54.21052631578947
IVL	53.684210526315	53.684210526315	54.73684210526319	53.157894736841	52.631578947368396	50.0	53.684210526315	54.21052631578947	54.73684210526319
KBANK	45.789473684210	47.368421052631	41.578947368421	46.842105263157904	46.315789473684	42.631578947368396	46.315789473684	46.842105263157904	42.105263157894704
KCE	58.9473684210526	59.473684210526	60.0	58.947368421052	59.473684210526	60.0	58.947368421052	59.473684210526	60.0
KTB	44.210526315789	43.684210526315795	41.578947368421	42.105263157894704	43.157894736842	43.684210526315795	42.631578947368396	43.684210526315795	41.578947368421
LH	43.1578947368421	38.947368421052	52.631578947368396	40.526315789473737	39.473684210526	40.526315789473737	43.1578947368421	38.947368421052	52.631578947368396
MINT	58.421052631578995	59.473684210526	56.842105263157904	57.894736842105296	59.473684210526	57.894736842105296	58.421052631578995	59.473684210526	56.842105263157904
MTLS	51.020408163265294	57.142857142857096	51.020408163265294	57.142857142857096	55.102040816326	55.102040816326	51.020408163265294	57.142857142857096	51.020408163265294
PS	54.736842105263	53.157894736842	63.157894736842	54.2105263157895	54.736842105263	62.631578947368	55.789473684210	53.157894736842	63.157894736842
PTT	40.0	41.578947368421	45.263157894736	38.421052631579	42.631578947368396	41.052631578974	40.526315789473737	41.578947368421	45.263157894736
PTTEP	46.842105263157904	47.368421052631	46.315789473684	46.842105263157904	47.368421052631	46.842105263157904	46.842105263157904	47.368421052631	46.315789473684
PTTGC	51.052631578947405	50.0	46.842105263157904	53.684210526315	49.473684210263	47.89473684210553	50.526315789473737	51.578947368421105	46.842105263157904
ROBINS	49.473684210263	46.315789473684	49.473684210263	48.421052631578895	44.736842105263	51.052631578947405	50.0	46.315789473684	48.947368421052
SAWAD	53.333333333333	52.222222222222	55.555555555556	51.111111111109	50.0	54.444444444439	52.222222222222	52.222222222222	55.555555555556
SCB	41.578947368421	41.052631578974	46.315789473684	41.052631578974	41.052631578974	44.736842105263	41.578947368421	41.578947368421	46.315789473684
SCC	51.052631578947405	52.631578947368396	52.105263157894	51.052631578947405	53.157894736842	53.684210526315	51.578947368421105	52.631578947368396	51.052631578947405
TASCO	53.684210526315	53.684210526315	50.0	53.684210526315	53.684210526315	52.105263157894	53.684210526315	53.684210526315	50.526315789473737
TCAP	55.789473684210	54.736842105263	52.105263157894	54.2105263157895	52.105263157894	52.631578947368396	55.789473684210	54.736842105263	52.105263157894
TMB	53.157894736842	53.157894736842	48.421052631578895	50.526315789473737	50.526315789473737	54.736842105263	52.105263157894	51.578947368421105	48.947368421052
TOP	58.9473684210526	59.473684210263	58.947368421052	58.947368421052	58.947368421052	58.947368421052	59.473684210263	59.473684210263	58.947368421052
TPIPL	58.421052631578995	56.842105263157904	57.368421052631604	58.947368421052	57.894736842105296	57.894736842105296	60.0	57.368421052631604	55.789473684210
TRUE	55.263157894736	52.105263157894	52.105263157894	53.684210526315	52.105263157894	52.631578947368396	55.263157894736	51.578947368421105	52.631578947368396
TTW	55.789473684210	58.947368421052	44.736842105263	55.263157894736	58.421052631578995	51.578947368421105	55.789473684210	57.368421052631604	45.263157894736
TU	50.0	51.052631578947405	47.894736842105	49.473684210263	51.052631578947405	50.526315789473737	50.0	51.052631578947405	47.894736842105
WHA	52.631578947368396	57.894736842105296	52.631578947368396	60.0	56.842105263157904	56.315789473684	52.631578947368396	58.947368421052	52.631578947368396

Table 121 MAPE results from closing change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	47.36842105263158	43.1578947368421	53.1578947368421	47.8947368421053	47.8947368421053	52.10526315789474	46.31578947368421	44.21052631578947	53.1578947368421
AOT	54.21052631578947	52.10526315789474	51.57894736842105	45.7894736842105	45.26315789473684	44.21052631578947	51.57894736842105	52.10526315789474	51.57894736842105
BA	29.629629629629626	38.88888888888889	33.33333333333333	55.55555555555556	35.185185185185205	59.25925925925926	31.48148148148148	38.88888888888889	33.33333333333333
BANPU	45.26315789473684	46.31578947368421	43.684210526315795	46.31578947368421	50.52631578947368	45.26315789473684	43.684210526315795	46.31578947368421	43.684210526315795
BBL	42.631578947368396	44.7368421052632	42.631578947368396	46.842105263157904	46.31578947368421	47.8947368421053	43.1578947368421	44.21052631578947	43.1578947368421
BCP	38.94736842105263	37.894736842105296	44.21052631578947	45.7894736842105	40.52631578947368	45.7894736842105	37.894736842105296	36.84210526315789	43.1578947368421
BDMS	47.8947368421053	50.0	48.94736842105263	45.7894736842105	41.578947368421105	48.421052631578895	47.8947368421053	46.842105263157904	48.94736842105263
BEC	48.421052631578895	52.10526315789474	49.473684210526316	48.94736842105263	46.842105263157904	49.473684210526316	48.421052631578895	52.10526315789474	49.473684210526316
BH	53.1578947368421	55.78947368421053	51.578947368421105	52.10526315789474	49.473684210526316	48.94736842105263	54.736842105263158	56.31578947368421	51.578947368421105
BLA	55.263157894736836	53.1578947368421	56.31578947368421	52.10526315789474	56.842105263157904	56.31578947368421	53.68421052631578	52.631578947368396	56.31578947368421
BTS	49.473684210526316	46.31578947368421	56.842105263157904	42.631578947368396	41.052631578947474	47.8947368421053	48.94736842105263	46.31578947368421	55.78947368421053
CBG	54.0	68.0	54.0	46.0	60.0	46.0	54.0	68.0	54.0
CENTEL	49.473684210526316	43.684210526315795	53.68421052631578	47.8947368421053	45.26315789473684	46.31578947368421	50.0	43.1578947368421	53.68421052631578
CK	48.94736842105263	45.26315789473684	48.94736842105263	44.210526315789	44.736842105263	47.894736842105	50.0	45.789473684210526	48.94736842105263
CPALL	51.052631578947405	55.263157894736836	52.10526315789474	48.421052631578895	51.578947368421105	48.421052631578895	51.052631578947405	55.263157894736836	52.10526315789474
CPF	43.684210526315795	48.421052631578895	47.8947368421053	46.31578947368421	54.210526315789405	44.736842105263246	44.210526315789405	48.421052631578895	47.8947368421053
CPN	45.26315789473684	55.78947368421053	47.8947368421053	43.684210526315795	38.94736842105263	44.210526315789405	45.26315789473684	55.78947368421053	47.8947368421053
DELTA	45.26315789473684	44.210526315789405	46.842105263157904	45.263157894736836	47.8947368421053	46.31578947368421	45.26315789473684	44.736842105263246	46.31578947368421
DTAC	52.10526315789474	46.31578947368421	48.421052631578895	48.421052631578895	47.8947368421053	46.31578947368421	53.15789473684211	46.31578947368421	48.421052631578895
EGCO	41.57894736842105	43.684210526315795	44.736842105263246	42.105263157894704	44.736842105263246	43.684210526315795	41.052631578947474	45.789473684210526	44.736842105263246
GLOW	46.31578947368421	42.631578947368396	48.421052631578895	45.789473684210526	45.789473684210526	48.421052631578895	47.368421052631576	43.684210526315795	47.8947368421053
GPSC	43.75	56.25	43.75	43.75	43.75	43.75	37.5	56.25	37.5
HMPRO	57.368421052631604	56.842105263157904	52.10526315789474	47.8947368421053	46.842105263157904	45.26315789473684	57.368421052631604	56.842105263157904	52.10526315789474
INTUCH	41.57894736842111	43.684210526315795	48.94736842105263	44.736842105263246	43.157894736842105	52.10526315789474	41.57894736842111	43.684210526315795	48.94736842105263
IRPC	48.421052631578895	51.052631578947405	51.052631578947405	51.052631578947405	52.10526315789474	52.631578947368396	50.52631578947368	50.52631578947368	51.052631578947405
IVL	54.736842105263158	55.263157894736836	55.263157894736836	47.368421052631576	45.26315789473684	47.8947368421053	55.789473684210526	55.263157894736836	55.263157894736836
KBANK	53.15789473684211	48.94736842105263	55.263157894736836	45.789473684210526	43.157894736842105	46.31578947368421	53.684210526315788	49.473684210526316	55.263157894736836
KCE	54.21052631578947	50.0	53.684210526315788	52.10526315789474	47.368421052631576	50.52631578947368	54.21052631578947	50.0	53.15789473684211
KTB	51.052631578947405	51.052631578947405	45.263157894736836	43.157894736842105	43.157894736842105	41.57894736842111	50.0	48.94736842105263	45.789473684210526
LH	46.842105263157904	49.473684210526316	45.789473684210526	38.421052631578904	43.157894736842105	40.0	47.368421052631576	50.0	45.789473684210526
MINT	44.7368421052632	43.684210526315795	50.52631578947368	46.31578947368421	46.31578947368421	51.578947368421105	44.210526315789405	42.105263157894704	50.52631578947368
MTLS	51.020408163265294	44.89795918367535	46.93877551020408	46.93877551020408	46.93877551020408	51.020408163265294	46.93877551020408	44.89795918367535	46.93877551020408
PS	50.52631578947368	46.31578947368421	49.473684210526316	44.736842105263246	48.421052631578895	45.26315789473684	51.052631578947405	45.263157894736836	49.473684210526316
PTT	51.578947368421105	49.473684210526316	51.052631578947405	41.05263157894737	40.52631578947368	47.368421052631576	53.15789473684211	49.473684210526316	51.052631578947405
PTTEP	51.052631578947405	53.684210526315788	48.421052631578895	43.684210526315795	42.631578947368396	42.105263157894704	48.94736842105263	52.631578947368396	48.421052631578895
PTTGC	52.631578947368396	53.684210526315788	54.21052631578947	57.894736842105296	58.94736842105263	58.421052631578995	52.631578947368396	53.15789473684211	54.736842105263194
ROBINS	54.736842105263194	55.263157894736836	53.684210526315788	40.0	60.0	37.894736842105296	54.736842105263194	55.789473684210526	53.684210526315788
SAWAD	53.333333333333336	47.77777777777778	47.77777777777778	40.0	38.88888888888889	41.11111111111111	52.22222222222222	48.88888888888889	47.77777777777778
SCB	50.52631578947368	49.473684210526316	52.10526315789474	47.368421052631576	54.736842105263194	56.31578947368421	49.473684210526316	49.473684210526316	51.578947368421105
SCC	45.789473684210526	46.842105263157904	43.684210526315795	46.31578947368421	42.631578947368396	42.631578947368396	45.263157894736836	45.263157894736836	43.684210526315795
TASCO	46.842105263157904	48.94736842105263	52.631578947368396	45.7894736842105	45.7894736842105	47.368421052631576	48.421052631578895	47.368421052631576	52.631578947368396
TCAP	50.0	51.052631578947405	50.52631578947368	52.10526315789474	54.210526315789405	55.263157894736836	48.421052631578895	52.631578947368396	50.0
TMB	42.631578947368396	41.05263157894737	43.157894736842105	49.473684210526316	50.52631578947368	48.94736842105263	44.210526315789405	41.57894736842111	43.157894736842105
TOP	52.10526315789474	47.368421052631576	48.94736842105263	53.684210526315788	51.052631578947405	54.210526315789405	51.052631578947405	49.473684210526316	49.473684210526316
TPIPL	47.368421052631576	46.31578947368421	46.31578947368421	47.368421052631576	47.368421052631576	49.473684210526316	46.842105263157904	45.789473684210526	46.31578947368421
TRUE	44.7368421052632	42.631578947368396	50.52631578947368	50.0	48.94736842105263	47.368421052631576	45.789473684210526	42.105263157894704	51.052631578947405
TTW	42.631578947368396	40.0	44.736842105263246	50.0	48.94736842105263	47.8947368421053	41.57894736842111	40.0	44.736842105263246
TU	45.789473684210526	44.210526315789405	41.57894736842111	40.0	43.684210526315795	37.894736842105296	46.31578947368421	43.157894736842105	41.57894736842111
WHA	48.94736842105263	45.789473684210526	44.210526315789405	52.631578947368396	53.15789473684211	54.736842105263194	48.421052631578895	44.210526315789405	43.684210526315795

Table 122 MAPE results from tick change.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	51.57894736842105	45.7894736842105	53.6842105263158	47.36842105263158	46.842105263157904	51.57894736842105	51.57894736842105	46.842105263157904	53.6842105263158
AOT	51.57894736842105	52.10526315789474	52.631578947368396	45.7894736842105	43.684210526315795	41.57894736842111	51.57894736842105	52.631578947368396	52.631578947368396
BA	33.33333333333333	38.88888888888889	33.33333333333333	35.185185185185205	33.33333333333333	35.185185185185205	31.48148148148148	38.88888888888889	33.33333333333333
BANPU	45.26315789473684	46.31578947368421	43.684210526315795	46.842105263157904	51.57894736842105	45.78947368421105	45.78947368421105	46.31578947368421	43.684210526315795
BBL	44.21052631578947	42.631578947368396	41.57894736842111	45.7894736842105	44.736842105263246	44.210526315789405	43.684210526315795	42.631578947368396	41.57894736842111
BCP	38.94736842105263	36.842105263157894	44.21052631578947	45.263157894736836	40.52631578947368	45.78947368421105	38.94736842105263	37.368421052631604	43.157894736842105
BDMS	47.36842105263158	50.526315789473684	48.94736842105263	45.7894736842105	43.157894736842105	48.421052631578895	46.842105263157904	50.526315789473684	48.94736842105263
BEC	49.473684210526316	52.631578947368396	50.0	47.368421052631576	47.368421052631576	49.473684210526316	49.473684210526316	52.631578947368396	50.0
BH	52.10526315789474	51.052631578947405	52.10526315789474	53.6842105263158	50.0	51.578947368421105	51.052631578947405	51.578947368421105	52.10526315789474
BLA	55.263157894736836	53.1578947368421	56.31578947368421	52.10526315789474	57.368421052631604	56.31578947368421	54.21052631578947	53.1578947368421	56.842105263157904
BTS	49.473684210526316	44.21052631578947	53.6842105263158	42.631578947368396	41.052631578947474	42.105263157894704	49.473684210526316	43.684210526315795	52.631578947368396
CBG	57.99999999999999	68.0	54.0	46.0	64.0	46.0	50.0	68.0	54.0
CENTEL	52.10526315789474	45.7894736842105	52.631578947368396	48.421052631578895	45.26315789473684	47.8947368421053	49.473684210526316	44.736842105263246	52.631578947368396
CK	46.842105263157904	47.894736842105296	50.526315789473684	44.736842105263246	45.789473684210526	44.736842105263246	46.842105263157904	48.421052631578895	50.526315789473684
CPALL	52.631578947368396	55.263157894736836	52.10526315789474	47.368421052631576	51.578947368421105	48.421052631578895	53.157894736842105	55.263157894736836	52.10526315789474
CPF	47.36842105263158	49.473684210526316	52.631578947368396	49.473684210526316	55.263157894736836	48.94736842105263	46.842105263157904	50.0	52.631578947368396
CPN	43.157894736842105	55.789473684210526	47.894736842105296	42.105263157894704	38.94736842105263	44.210526315789405	47.894736842105296	55.789473684210526	47.894736842105263
DELTA	45.789473684210526	44.736842105263246	46.842105263157904	46.315789473684246	47.894736842105263	46.315789473684246	45.263157894736836	44.736842105263246	46.315789473684246
DTAC	48.421052631578895	48.421052631578895	49.473684210526316	46.842105263157904	47.894736842105263	48.421052631578895	50.0	44.210526315789405	49.473684210526316
EGCO	42.105263157894704	45.789473684210526	44.736842105263246	43.157894736842105	43.157894736842105	43.684210526315795	42.631578947368396	45.789473684210526	44.736842105263246
GLOW	46.842105263157904	42.631578947368396	46.315789473684246	43.157894736842105	45.789473684210526	47.894736842105263	46.842105263157904	44.210526315789405	46.315789473684246
GPSC	43.75	56.25	43.75	43.75	43.75	43.75	43.75	56.25	37.5
HMPRO	57.368421052631604	56.842105263157904	52.631578947368396	48.421052631578895	46.842105263157904	46.315789473684246	56.315789473684246	57.368421052631604	53.157894736842105
INTUCH	41.052631578947474	43.684210526315795	48.94736842105263	44.736842105263246	43.157894736842105	52.10526315789474	41.052631578947474	44.736842105263246	48.94736842105263
IRPC	48.94736842105263	50.526315789473684	51.052631578947405	50.526315789473684	49.473684210526316	54.210526315789405	48.94736842105263	52.105263157894704	51.052631578947405
IVL	52.631578947368396	53.684210526315788	54.21052631578947	49.473684210526316	48.94736842105263	50.0	53.684210526315788	53.684210526315788	53.684210526315788
KBANK	52.631578947368396	50.526315789473684	54.736842105263194	51.578947368421105	51.052631578947405	56.842105263157904	53.684210526315788	48.94736842105263	54.210526315789405
KCE	54.21052631578947	51.052631578947405	51.578947368421105	53.6842105263158	45.263157894736836	52.631578947368396	54.736842105263194	50.0	51.578947368421105
KTB	51.052631578947405	51.052631578947405	45.263157894736836	42.631578947368396	42.631578947368396	41.57894736842111	49.473684210526316	51.052631578947405	45.789473684210526
LH	45.26315789473684	49.473684210526316	41.57894736842111	47.8947368421053	48.421052631578895	51.578947368421105	44.210526315789405	48.421052631578895	41.57894736842111
MINT	45.789473684210526	42.631578947368396	47.894736842105263	47.368421052631576	48.94736842105263	50.0	46.315789473684246	42.631578947368396	47.894736842105263
MTLS	51.020408163265294	44.89795918367535	46.93877551020408	46.97959183673469	46.93877551020408	51.020408163265294	51.020408163265294	44.89795918367535	46.93877551020408
PS	46.31578947368421	44.736842105263246	47.894736842105263	45.263157894736836	44.210526315789405	46.31578947368421	44.736842105263246	44.736842105263246	48.94736842105263
PTT	51.052631578947405	48.94736842105263	51.052631578947405	40.52631578947368	40.52631578947368	47.368421052631576	50.526315789473684	48.421052631578895	51.052631578947405
PTTEP	53.15789473684211	52.631578947368396	52.10526315789474	42.631578947368396	42.631578947368396	39.473684210526263	53.15789473684211	52.631578947368396	52.631578947368396
PTTGC	54.21052631578947	53.15789473684211	54.21052631578947	58.421052631578995	58.94736842105263	58.421052631578995	53.684210526315788	54.21052631578947	54.736842105263194
ROBINS	54.736842105263194	56.31578947368421	53.684210526315788	40.52631578947368	61.052631578947405	37.894736842105296	54.21052631578947	56.842105263157904	53.684210526315788
SAWAD	52.22222222222222	48.88888888888889	47.77777777777778	41.11111111111111	40.0	41.11111111111111	53.333333333333336	47.77777777777778	46.666666666666664
SCB	50.52631578947368	48.94736842105263	52.10526315789474	56.842105263157904	54.21052631578947	56.31578947368421	50.0	50.0	51.578947368421105
SCC	46.842105263157904	44.736842105263246	43.157894736842105	45.263157894736836	42.631578947368396	43.157894736842105	46.842105263157904	45.263157894736836	43.157894736842105
TASCO	51.052631578947405	48.421052631578895	49.473684210526316	49.473684210526316	48.421052631578895	49.473684210526316	51.052631578947405	48.421052631578895	49.473684210526316
TCAP	50.0	52.10526315789474	50.526315789473684	52.10526315789474	54.21052631578947	55.263157894736836	47.894736842105263	52.631578947368396	50.0
TMB	43.684210526315795	42.631578947368396	43.157894736842105	48.94736842105263	50.52631578947368	48.94736842105263	42.631578947368396	41.57894736842111	43.684210526315795
TOP	51.052631578947405	48.421052631578895	48.947368421052526	53.684210526315788	51.052631578947405	54.210526315789405	47.368421052631576	46.842105263157904	49.473684210526316
TPIPL	51.052631578947405	48.421052631578895	50.526315789473684	47.894736842105263	46.315789473684246	47.894736842105263	48.94736842105263	48.94736842105263	50.526315789473684
TRUE	44.736842105263246	44.736842105263246	49.473684210526316	45.263157894736836	44.210526315789405	46.842105263157904	46.315789473684246	45.789473684210526	49.473684210526316
TTW	41.05263157894737	40.0	42.631578947368396	40.0	39.473684210526263	46.315789473684246	42.105263157894704	38.94736842105263	42.105263157894704
TU	45.789473684210526	44.736842105263246	41.57894736842111	41.57894736842111	44.210526315789405	37.894736842105296	44.736842105263246	43.157894736842105	41.57894736842111
WHA	50.526315789473684	44.736842105263246	43.157894736842105	53.157894736842105	53.684210526315788	51.052631578947405	49.473684210526316	44.736842105263246	43.157894736842105

Table 123 MAPE results from direction.

Stock	eps-svr_rbfdot	eps-svr_laplacedot	eps-svr_besseldot	nu-svr_rbfdot	nu-svr_laplacedot	nu-svr_besseldot	eps-bsvr_rbfdot	eps-bsvr_laplacedot	eps-bsvr_besseldot
ADVANC	45.7894736842105	43.684210526315795	45.2631578947368	40.5263157894737	43.684210526315795	55.26315789473684	45.7894736842105	43.684210526315795	45.26315789473684
AOT	39.4736842105263	36.31578947368421	41.052631578947	47.8947368421053	44.2105263157895	50.52631578947368	39.4736842105263	35.26315789473684	41.57894736842111
BA	35.185185185185205	38.88888888888889	35.185185185185205	42.592592592592595	33.33333333333333	48.148148148148	38.88888888888889	38.88888888888889	35.185185185185205
BANPU	42.631578947368396	41.57894736842111	40.0	46.31578947368421	46.31578947368421	45.78947368421053	41.57894736842111	41.57894736842111	40.0
BBL	39.4736842105263	42.105263157894704	42.631578947368396	45.78947368421053	46.31578947368421	54.736842105263194	40.52631578947368	42.105263157894704	42.105263157894704
BCP	39.4736842105263	38.94736842105263	40.0	41.57894736842111	38.42105263157895	51.578947368421105	39.4736842105263	38.94736842105263	40.0
BDMS	38.94736842105263	37.368421052631604	37.894736842105296	46.31578947368421	39.473684210526316	50.52631578947368	39.4736842105263	37.368421052631604	37.894736842105296
BEC	48.421052631578895	41.57894736842111	48.421052631578895	47.368421052631576	48.421052631578895	45.26315789473684	47.368421052631576	43.15789473684211	48.421052631578895
BH	53.15789473684211	49.4736842105263	56.842105263157904	52.631578947368396	55.78947368421053	53.68421052631579	54.21052631578947	49.4736842105263	56.842105263157904
BLA	47.368421052631576	45.78947368421053	50.0	51.578947368421105	52.10526315789474	55.26315789473684	47.894736842105263	45.78947368421053	49.473684210526316
BTS	43.684210526315795	41.05263157894737	42.631578947368396	40.52631578947368	42.631578947368396	55.26315789473684	43.684210526315795	41.05263157894737	42.105263157894704
CBG	44.0	32.0	46.0	34.0	66.0	38.0	44.0	32.0	48.0
CENTEL	39.4736842105263	37.894736842105296	40.0	49.473684210526316	45.26315789473684	55.26315789473684	38.94736842105263	38.42105263157895	40.0
CK	38.94736842105263	38.94736842105263	38.947368421052636	47.894736842105263	45.26315789473684	52.10526315789474	37.894736842105296	38.94736842105263	38.947368421052636
CPALL	44.736842105263216	43.15789473684211	44.736842105263216	45.26315789473684	46.31578947368421	46.31578947368421	44.736842105263216	43.15789473684211	43.684210526315795
CPF	48.421052631578895	43.684210526315795	49.473684210526316	54.736842105263194	51.052631578947405	45.26315789473684	46.31578947368421	43.684210526315795	49.473684210526316
CPN	42.105263157894704	46.31578947368421	37.368421052631604	50.52631578947368	50.52631578947368	44.736842105263216	42.105263157894704	46.31578947368421	37.368421052631604
DELTA	45.78947368421053	45.26315789473684	48.421052631578895	41.57894736842111	42.631578947368396	50.52631578947368	45.78947368421053	45.26315789473684	49.473684210526316
DTAC	45.78947368421053	51.052631578947405	47.894736842105263	55.78947368421053	51.578947368421105	51.578947368421105	46.31578947368421	51.052631578947405	47.894736842105263
EGCO	41.05263157894737	41.57894736842111	38.947368421052636	42.631578947368396	39.473684210526316	47.894736842105263	39.4736842105263	40.0	38.947368421052636
GLOW	40.52631578947368	40.0	43.15789473684211	47.894736842105263	45.26315789473684	51.052631578947405	41.57894736842111	38.94736842105263	43.15789473684211
GPSC	43.75	43.75	43.75	50.0	37.5	43.75	43.75	43.75	43.75
HMPRO	40.0	36.84210526315789	36.31578947368421	49.473684210526316	52.631578947368396	37.894736842105296	36.84210526315789	36.84210526315789	36.31578947368421
INTUCH	43.15789473684211	42.105263157894704	42.105263157894704	44.2105263157895	42.105263157894704	53.68421052631579	43.15789473684211	42.105263157894704	42.105263157894704
IRPC	43.684210526315795	42.105263157894704	43.15789473684211	51.052631578947405	50.0	45.26315789473684	42.631578947368396	42.105263157894704	43.684210526315795
IVL	43.15789473684211	38.94736842105263	40.52631578947368	48.421052631578895	48.421052631578895	50.0	43.15789473684211	38.94736842105263	40.52631578947368
KBANK	43.15789473684211	44.21052631578947	43.684210526315795	60.526315789473694	54.21052631578947	52.631578947368396	43.15789473684211	43.15789473684211	43.15789473684211
KCE	42.105263157894704	40.0	40.52631578947368	46.31578947368421	51.052631578947405	45.78947368421053	40.0	40.0	40.52631578947368
KTB	39.4736842105263	39.473684210526316	39.473684210526316	50.52631578947368	46.31578947368421	42.631578947368396	39.4736842105263	39.473684210526316	39.473684210526316
LH	38.42105263157895	38.42105263157895	38.421052631578946	41.052631578947405	40.52631578947368	49.473684210526316	38.42105263157895	38.42105263157895	38.42105263157895
MINT	37.894736842105296	39.473684210526316	41.052631578947405	43.684210526315795	38.94736842105263	50.52631578947368	38.94736842105263	38.94736842105263	40.52631578947368
MTLS	48.9795918367347	48.9795918367347	46.938775510204104	46.938775510204104	53.06122448979592	51.020408163265294	46.938775510204104	48.9795918367347	46.938775510204104
PS	37.894736842105296	36.31578947368421	37.368421052631604	52.631578947368396	47.36842105263158	37.894736842105296	37.894736842105296	36.31578947368421	37.894736842105296
PTT	44.21052631578947	44.736842105263216	46.31578947368421	47.8947368421053	46.842105263157904	50.52631578947368	44.736842105263216	44.736842105263216	46.842105263157904
PTTEP	53.15789473684211	54.736842105263194	55.26315789473684	47.8947368421053	50.52631578947368	52.10526315789474	53.15789473684211	54.736842105263194	55.26315789473684
PTTGC	45.78947368421053	48.94736842105263	46.31578947368421	53.15789473684211	49.473684210526316	46.842105263157904	45.78947368421053	50.0	46.31578947368421
ROBINS	46.31578947368421	42.105263157894704	46.31578947368421	42.631578947368396	48.421052631578895	41.57894736842111	46.842105263157904	43.684210526315795	46.842105263157904
SAWAD	45.55555555555556	41.11111111111111	50.0	46.666666666666714	50.0	57.77777777777779	43.333333333333336	41.11111111111111	47.77777777777778
SCB	50.52631578947368	44.21052631578947	48.94736842105263	53.15789473684211	50.52631578947368	45.78947368421053	47.36842105263158	43.684210526315795	48.94736842105263
SCC	38.94736842105263	41.57894736842111	42.105263157894704	48.421052631578895	50.52631578947368	47.894736842105263	38.42105263157895	41.57894736842111	42.105263157894704
TASCO	45.26315789473684	46.31578947368421	43.684210526315795	52.10526315789474	44.736842105263216	56.842105263157904	45.26315789473684	46.31578947368421	43.15789473684211
TCAP	37.368421052631604	38.42105263157895	38.42105263157895	47.368421052631576	48.421052631578895	50.0	38.42105263157895	38.42105263157895	38.42105263157895
TMB	43.684210526315795	43.684210526315795	43.684210526315795	43.684210526315795	44.736842105263216	47.368421052631576	43.684210526315795	43.684210526315795	43.684210526315795
TOP	43.684210526315795	42.105263157894704	41.57894736842111	47.894736842105263	45.78947368421053	46.842105263157904	43.15789473684211	42.631578947368396	41.57894736842111
TPIPL	44.736842105263216	43.15789473684211	43.15789473684211	46.31578947368421	50.52631578947368	53.68421052631579	42.631578947368396	43.15789473684211	43.15789473684211
TRUE	45.26315789473684	44.21052631578947	43.15789473684211	48.94736842105263	41.052631578947405	53.15789473684211	43.684210526315795	44.21052631578947	42.105263157894704
TTW	34.21052631578947	34.736842105263216	34.21052631578947	46.31578947368421	42.631578947368396	55.26315789473684	33.68421052631579	34.736842105263216	34.21052631578947
TU	37.894736842105296	38.42105263157895	37.894736842105296	55.78947368421053	52.10526315789474	54.21052631578947	37.894736842105296	38.42105263157895	37.894736842105296
WHA	40.0	38.42105263157895	39.473684210526316	44.2105263157895	43.684210526315795	49.473684210526316	40.0	37.368421052631604	39.473684210526316

CURRICULUM VITAE

NAME : Piyatat Chatvorawit **GENDER :** Male

DATE OF BIRTH : October 23, 1984 **NATIONALITY :** Thai

EDUCATION BACKGROUND :

- Bachelor of Science in Mathematics (First Class Honors), Khonkaen University, Thailand, 2007
- Master of Science in Computer Science, Asian Institute of Technology, Thailand, 2009

SCHOLARSHIP :

- Development and Promotion of Science and Technology Talents Project (DPST), 2000-2007
- Royal Thai Government (RTG) Fellowships, 2007-2009

PUBLICATION :

- Improving Stock Price Prediction with SVM by Simple Transformation: The Sample of Stock Exchange of Thailand (SET), **Thai Journal of Mathematics**