DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION

NISACHON TABCHA

มหาวิทยาลัยเทคโนโลยีสุรนารี

A Thesis Submitted in Partial Fulfillment of the Requirements for
the Degree of Master of Engineering in Electrical Engineering
Suranaree University of Technology
Academic Year 2025

การพยากรณ์ภาระโหลดล่วงหน้ารายวันโดยใช้เทคนิคการเรียนรู้เชิงลึกแบบมีผู้สอนร่วมกับวิศวกรรมคุณลักษณะเชิงบริบทและการวิเคราะห์แบบอธิบายได้: กรณีศึกษามหาวิทยาลัยเทคโนโลยีสุรนารีที่มีการบูรณาการพลังงานแสงอาทิตย์

นางสาวนิศาชล ทับชา

มหาวิทยาลัยเทคโนโลยีสุรนารี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2568

DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

Somchat Jiriwibhakorn
.....

(Assoc. Prof. Dr. Somchat Jiriwibhakorn)

Chairperson

.....

(Assoc. Prof. Dr. Keerati Chayakulkheeree)

Member (Thesis Advisor)

.....

(Asst. Prof. Dr. Nitikorn Junhuathon)

Member (Thesis Co-Advisor)

.....

(Prof. Dr. Thanatchai Kulworawanichpong)

Member

Tosaphol Ratniyomchai
.....

(Asst. Prof. Dr. Tosaphol Ratniyomchai)

Member

Yupaporn
.....

(Assoc. Prof. Dr. Yupaporn Ruksakulpiwat)

Acting Vice Rector for Academic Affairs and Quality Assurance

.....

(Assoc. Prof. Dr. Pornsiri Jongkol)

Dean of Institute of Engineering

มหาวิทยาลัยเทคโนโลยีสุรนารี

นิศาชล ทับชา : การพยากรณ์ภาระโหลดล่วงหน้ารายวันโดยใช้เทคนิคการเรียนรู้เชิงลึกแบบมีผู้สอนร่วมกับวิศวกรรมคุณลักษณะเชิงบริบทและการวิเคราะห์แบบอธิบายได้: กรณีศึกษามหาวิทยาลัยเทคโนโลยีสุรนารีที่มีการบูรณาการพลังงานแสงอาทิตย์ (DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร. กีรติ ชยะกุลคีรี, 131 หน้า.

คำสำคัญ : การพยากรณ์โหลดไฟฟ้า, สารสนเทศร่วม, ปัญญาประดิษฐ์ที่อธิบายได้, โครงข่ายประสาทเทียม, อธิบายโมเดลด้วยค่าแชปลีย์

งานวิจัยนี้นำเสนอกรอบการทำงานสำหรับการพยากรณ์โหลดไฟฟ้าระยะสั้น และประเมินประสิทธิภาพของแบบจำลองเชิงลึกโดยใช้ข้อมูลโหลดไฟฟ้าที่มีความละเอียดเชิงเวลา 15 นาที แบบจำลองที่พัฒนาขึ้นใช้ฟีเจอร์ด้านเวลา โหลดไฟฟ้าย้อนหลัง และตัวแปรด้านสภาพอากาศ ร่วมกับกระบวนการสร้างและคัดเลือกฟีเจอร์ที่เหมาะสม แบบจำลองที่นำมาศึกษาเปรียบเทียบประกอบด้วย โครงข่ายประสาทเทียมเชิงลึก โครงข่ายประสาทเทียมหน่วยความจำระยะยาวแบบทางเดียวและสองทิศทาง โครงข่ายประสาทเทียมแบบคอนโวลูชันร่วมกับหน่วยความจำระยะยาว วิธีค่าเฉลี่ยเคลื่อนที่ และโครงข่ายประสาทเทียมที่ใช้กลไกความสนใจ โดยมีการพิจารณาแบบจำลองหน่วยความจำที่มีการควบคุมด้วยประตูแบบทางเดียวและสองทิศทางในการทดสอบเบื้องต้น

เพื่อเพิ่มความแม่นยำและความสามารถในการอธิบายแบบจำลอง งานวิจัยนี้ได้นำวิธีสารสนเทศร่วมและวิธีการอธิบายโมเดลด้วยค่าแชปลีย์มาใช้ในการคัดเลือกฟีเจอร์ การประเมินผลดำเนินการด้วยตัวชี้วัด ผลการทดลองแสดงให้เห็นว่าแบบจำลองที่สามารถเรียนรู้ความสัมพันธ์เชิงลำดับเวลาได้อย่างชัดเจน โดยเฉพาะแบบจำลองหน่วยความจำระยะยาวแบบสองทิศทางและแบบจำลองคอนโวลูชันร่วมกับหน่วยความจำระยะยาว ให้ผลการพยากรณ์ที่ดีกว่าแบบจำลองพื้นฐาน อีกทั้งการคัดเลือกฟีเจอร์ด้วยวิธีการอธิบายโมเดลด้วยค่าแชปลีย์ให้ผลลัพธ์ที่สอดคล้องกับกลไกการเรียนรู้ของแบบจำลองมากกว่าวิธีสารสนเทศร่วม

สาขาวิชาวิศวกรรมไฟฟ้า ลายมือชื่อนักศึกษา นิศาชล

ปีการศึกษา 2568 ลายมือชื่ออาจารย์ที่ปรึกษา

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

NISACHON THABCHA : DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION

THESIS ADVISOR : ASSOC. PROF. KEERATI CHAYAKULKHEEREE, Ph.D., 131 PP.

Keyword : Load Forecasting / Mutual Information / Explainable AI / Neural Network / SHAP

This study proposes a framework for short-term electric load forecasting and evaluates the performance of deep learning models using electric load data with a 15-minute resolution. The forecasting model incorporates time-related features, historical load data, and meteorological variables, along with feature engineering for effective feature construction and selection.

The evaluated models include Deep Neural Network (DNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Convolutional Neural Network–LSTM (CNN-LSTM), Moving Average, and Attention-based LSTM, with Gated Recurrent Unit (GRU) and Bidirectional GRU (Bi-GRU) considered in preliminary experiments.

To improve forecasting accuracy and model interpretability, Mutual Information (MI) and Shapley Additive Explanations (SHAP) are applied for feature selection. Model performance is assessed using MAE, RMSE, and MAPE. The results demonstrate that models explicitly capturing temporal dependencies, particularly Bi-LSTM and CNN-LSTM, outperform baseline models. Furthermore, SHAP-based feature selection provides feature subsets that better align with the learning mechanisms of deep learning models compared to MI.

มหาวิทยาลัยเทคโนโลยีสุรนารี

School of Electrical Engineering

Academic Year 2025

Student's Signature .....

Advisor's Signature .....

Co-Advisor's Signature .....

ACKNOWLEDGEMENT

The authors are particularly grateful to the Kittibandit Scholarship at Suranaree University of Technology, for which I am sincerely grateful.

I appreciate my advisor, Assoc. Prof. Dr. Keerati Chayakulkheeree, whose guidance, insightful feedback, and constant encouragement have been instrumental in the completion of this work. Beyond academic advice, I am deeply grateful for the valuable life skills and practical wisdom I have gained from him in my daily life. His mentorship has not only shaped my research abilities but also inspired me to develop discipline, perseverance, and a positive attitude in facing challenges both in my studies and in everyday living. I also extend my sincere thanks to the committee members: Assoc. Prof. Dr. Somchat Jiriwibhakorn, Assoc. Prof. Dr. Keerati Chayakulkheeree, Asst. Prof. Dr. Nitikorn Junhuathon, Prof. Dr. Thanatchai Kulworawanichpong, and Asst. Prof. Dr. Tosaphol Ratniyomchai, for their valuable advice, suggestions, and support throughout the thesis evaluation process

Special appreciation is given to my parents for their enduring support, both emotionally and financially. Their belief in me has been a cornerstone of my academic journey. I am deeply grateful for everything they have done for me.

Furthermore, I would like to acknowledge all my lecturers at the School of Electrical Engineering for providing me with foundational knowledge and academic mentorship during my studies. I am also grateful to my friends, my family, and seniors, whose collaboration, encouragement, and assistance greatly contributed to this work.

Lastly, I would like to thank the faculty members and administrative staff at Suranaree University of Technology for their support and kind assistance throughout my academic life.

Nisachon Thabcha

TABLE OF CONTENTS

Page

ABSTRACT (THAI) I
ABSTRACT (ENGLISH) II
ACKNOWLEDGEMENT III
TABLE OF CONTENTS IV
LIST OF TABLES VIII
LIST OF FIGURES X

CHAPTER

I INTRODUCTION 1

1.1 General Introduction 1
1.2 Problem Statement 2
1.3 Research Objectives 3
1.4 Scope and limitations 4
1.4.1 Scope 4
1.4.2 limitations 5
1.5 Conception 5
1.6 Research Benefits 6
1.7 Structure of thesis 6

II LITERATURE REVIEW 8

2.1 Introduction 8
2.2 Literature Overview 9
2.3 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting 14
2.4 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting 18
2.5 Research Gaps in Load Forecasting Framework 22

มหาวิทยาลัยเทคโนโลยีสุรนารี

TABLE OF CONTENTS (Continued)

Page

III Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangemen 24

3.1 Introduction 24
3.2 Data Collection and Preparation 25
3.3 Feature Extraction 26
3.4 Feature Engineering 26
3.4.1 Feature Transformation / Binarization 27
3.4.2 Time-based Feature Engineering / Ordinal Time Block 27
3.4.3 Time Component Extraction 28
3.4.4 Time Series Features / Lag Features 28
3.4.5 Ambient Temperature 28
3.4.6 Dew Point 29
3.5 Advanced RNN Architectures 29
3.5.1 LSTM Architecture 29
3.5.2 Bi-LSTM Architecture 32
3.5.3 GRU Architecture 32
3.5.4 Bi-GRU Architecture 32
3.6 Workflow of the Proposed Method 33
3.7 Performance Factor 34
3.8 Simulation Results 35
3.8.1 Experimental setup 35
3.8.2 Model performance in validation and model training 37
3.8.3 Performance of each model on weekday and weekend loads ... 39

มหาวิทยาลัยเทคโนโลยีสุรนารี

TABLE OF CONTENTS (Continued)

Page

3.8.4 Performance of LSTM model on 12 test sets 44
3.9 Summary of the daily forecast 47
IV Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus 48
4.1 Introduction 48
4.2 Data Collection and Preparation 48
4.3 Feature Modeling and Representation of Electrical Load Patterns 50
4.3.1 SUT's Campus Data 50
4.3.2 New England (ISO-NE) Data 50
4.4 Feature Selection and Mutual Information 51
4.5 Modeling Approaches and Baseline Methods 52
4.5.1 Attention-LSTM Model 53
4.5.2 CNN-LSTM Model 55
4.5.3 DNN Model 56
4.5.4 Moving average 56
4.6 Time Series Cross-Validation (TSCV) 57
4.7 Shapley Additive Explanation (SHAP) 58
4.8 Forecasting Process Description 59
4.9 Load Forecasting Framework 62
4.10 Simulation Results and Discussion with SUT's Campus Data 63
4.10.1 Implementation and Experimental Setup 63
4.10.2 Mutual Information Scores of SUT Data 66
4.10.3 Daily Load Forecasting Performance 67
4.10.4 Test Set Load Forecasting Performance 71
4.10.5 SHAP-Based Feature Importance Analysis 72

มหาวิทยาลัยเทคโนโลยีสุรนารี

TABLE OF CONTENTS (Continued)

Page

4.10.6 Comparative Analysis of Feature Importance Models and Methods....73

4.11 Simulation Result and Discussion with ISO New England Data....76

4.11.1 Mutual Information Scores of ISO-NE Data....76

4.11.2 Daily Load Forecasting Performance....77

4.11.3 Test Set Load Forecasting Performance....80

4.11.4 SHAP-Based Feature Importance Analysis....81

4.11.4 Comparative Analysis of Feature Importance Models and Methods....82

4.12 Summary of Results....84

V CONCLUSION....85

5.1 Conclusion....85

5.2 Suggestions....50

REFERENCES....93

APPENDIX A Performance of LSTM Model on 12 Test Sets....**94**

APPENDIX B Implementation and Experimental Setup....**96**

APPENDIX C Test Set Load Forecasting Performance....**101**

APPENDIX D SHAP-Based Feature Importance Analysis....**107**

APPENDIX E List of publications....**113**

BIOGRAPHY....131

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF TABLES

Table **Page**

2.1 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting....10
2.2 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research....12
3.1 Performance Factor 34
3.2 Model parameter settings 36
3.3 Evaluation results of model performance in validation metrics and model training time....37
3.4 Performance of each model on weekday load 42
3.5 Performance of each model on weekend load 42
3.6 Model Performance on Validation and Average Performance on 12 Test Sets 46
4.1 Parameter Setting 64
4.2 Selecting the appropriate hyperparameter....65
4.3 Details of parameter adjustment via indicators....65
4.4 Performance of Each Model on 10 Test Sets....67

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF TABLES (Continued)

Table **Page**

4.5 Overall Test Set Performance Metrics....72
4.6 Feature Importance Ranking of NN Model by SHAP and MI....74
4.7 The best result for each model in daily forecasting....78
4.8 Overall Test Set Performance Metrics of ISO-NE Data....81
4.9 Feature Importance Ranking of ISO-NE Data....83
A.1 Performance of LSTM Model on 12 Test Sets....94

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF FIGURES

Figure **Page**

3.1 LSTM architecture 30
3.2 Bi-LSTM architecture 31
3.3 GRU architecture 32
3.4 Bi-GRU architecture 33
3.5 The Proposed Load Forecasting Workflow 33
3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models.....39
3.7 One-day-ahead electricity load forecast for each model on weekdays.....41
3.8 One-day-ahead electricity load forecast for each model on weekends.....41
3.9 Forecasting Result – Test Set 3: Exam Period.....45
3.10 Forecasting Result – Test Set 6: Start of the Semester.....45
4.1 Preparation of Actual Electricity Load Data 49
4.2 Attention-LSTM architecture 54
4.3 Expanding Window.....57
4.4 Process Description 61
4.5 The Complete Forecasting Framework 63

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF FIGURES (Continued)

Figure **Page**

4.6 Mutual Information Scores of SUT Data.....66
4.7 The Lowest-Performing Forecast Results in the Initial Semester Test Set.....68
4.8 The Lowest-Performing Forecast Results in the Rainy Season Test Set.....68
4.9 The Best-Performing Forecasting Results in the Test Set during the Exam Period.....69
4.10 SHAP Value of Each Model: Bi-LSTM 73
4.11 Mutual Information Scores of ISO-NE Data 77
4.12 Comparison of the best forecasting results of the CNN-LSTM model.....79
4.13 SHAP Value of Each Model of ISO-NE Data: CNN-LSTM 82
C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM 101
C.2 Performance evaluation graph of the model's test suite: a) Bi-LSTM (Top 3 Features of SHAP) and b) Bi-LSTM (Top 3 Features of MI).....103
C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features 104
D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM 107
D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM 110

CHAPTER I

INTRODUCTION

1.1 General Introduction

From the Smart Grid Driving Plan, according to the framework of the Master Plan for the Development of Thailand's Smart Grid System 2015-2036, it will promote the development of various infrastructures and the management of resources in the power distribution system that are necessary to support the transition to a new generation of power grids efficiently and environmentally friendly. This will help support Thailand to move towards clean energy and reduce carbon dioxide emissions, as well as to support the energy transition, which requires the development of a flexible power system to support the increase in renewable energy in line with the national energy plan targets. This will be an option for maintaining the country's electricity balance and managing peak electricity demand efficiently, divided into 5 main pillars, one of which is the Load Response and Energy Management System (DR & EMS). In the past few years, the demand for energy has grown rapidly. The increasing use of renewable energy and the development of digital technology have changed the operation and planning of the power system. These changes have caused both utilities and energy consumers to adopt smarter and more demand-responsive solutions to manage energy more efficiently and sustainably.

Electricity usage forecasting, which is the ability to predict future energy use, plays an important role in achieving this goal. Especially in universities or commercial buildings where electricity usage fluctuates with academic calendars, holidays, seasons, and weather conditions. Accurate short-term forecasting is essential to effectively manage the demand side, schedule, and integrate with renewable energy sources. One case is Suranaree University of Technology (SUT), located in Nakhon Ratchasima, Thailand. The university has a photovoltaic (PV) solar system and is monitored by smart meters that report electricity usage every 15 minutes. However,

the presence of solar power makes it difficult to directly observe the actual electricity demand (net load) because the measured grid load is compensated for by solar power generation. This study reconstructs the pure load by combining the grid load and solar power generation to represent the university's actual energy demand, which serves as the target for one-day load forecasting.

This research utilizes deep learning models in the field of neural networks combined with advanced methods such as K-fold validation, Mutual Information (MI), and Shapley Additive Explanation (SHAP) interpretation to improve forecasting accuracy and clarity. The goal is not only to accurately forecast electricity consumption but also to enhance the intelligence and responsiveness of the university-level energy system in line with the direction of Thailand's national smart grid system.

1.2 Problem Statement

In an era where energy demand is highly volatile and renewable energy systems are continuously integrated into the power grid, accurate electricity load forecasting is crucial for efficient energy management, especially in educational institutions like Suranaree University of Technology (SUT), which has inconsistent energy usage patterns that vary depending on weekdays, public holidays, semesters, and weather conditions such as temperature and humidity that affect air conditioning usage. The solar PV system installed within the university further complicates energy measurement because the energy meter only reports the net load after deducting the energy generated by the solar cells. This results in missing data on actual energy usage, which makes forecasting using traditional methods such as linear regression or ARIMA inaccurate and unable to handle contextual complexity.

The problem statement is as follows,

1) Accurate electricity load forecasting is essential for efficient energy management, especially in systems integrated with solar PV. However, existing models often rely on incomplete load data derived from subtracting PV generation from metered values, resulting in inaccurate representations of actual energy consumption. Additionally, traditional forecasting methods struggle to capture the complex, context-

dependent nature of energy usage patterns influenced by time, weather, and behavioral variations.

2) Improving the accuracy of electricity load forecasting requires efficient feature engineering. Because energy consumption behavior data is complex and changes with contexts such as time, weather, weekdays, or holidays, which have non-linear relationships with electricity load. However, traditional methods using linear techniques such as correlation analysis or feature selection based on statistical values often cannot capture these deep or hidden relationships. As a result, the resulting models have low performance and cannot accurately reflect energy consumption behavior in real situations or do not cover important contexts.

3) The transparency of deep learning models is also a limitation in practical applications. Although deep learning models can provide accurate predictions, they have limitations in interpreting and explaining the results. They cannot explain models with the characteristics of a black box, resulting in a lack of transparency and an inability to effectively support decision-making.

1.3 Research Objectives

To address such challenges, this study aims to develop a novel short-term electricity load forecasting model using deep learning techniques in the RNN family and enhance its accuracy with contextual feature engineering and interpretability tools to enable more accurate, transparent, and efficient energy decision-making for short-term (one-day-ahead) electricity load forecasting.

The main objectives of this thesis are as follows,

1) To develop a one-day electricity load forecasting model using deep learning techniques. We will compare deep learning models with basic models, including Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. Specifically, we will examine the model with the highest accuracy based on real-world load data reconstructed from electricity meters and solar panels.

2) To improve accuracy and reliability through the design of appropriate and high-quality features. Mutual Information (MI) is a promising approach for capturing

complex nonlinear energy consumption patterns. and systematic evaluation using performance indicators including RMSE, MAE, and MAPE.

3) To improve the transparency and interpretability of the model, SHAP analysis is used to identify influential features and analyze their importance to gain a deeper understanding of the model's performance.

1.4 Scope and limitations

1.4.1 Scope

This study focuses on electrical load forecasting using deep learning techniques. In addition, it studies the use of feature engineering techniques to improve the input data for the learning model, including the selection of necessary features and the description of variables through the model. In summary, the scopes are considered as follows:

1) This study focuses on the development of a one-day-ahead electricity load forecasting model for Suranaree University of Technology (SUT) using deep learning techniques including moving average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM.

2) The target variable used in the forecasting model is the pure load, reconstructed by combining grid-based electricity consumption with solar PV generation data to represent the university's actual energy demand.

3) The input features are designed to reflect real-world contexts, including temporal data (e.g., yearly hours, quarter-hour intervals), calendar effects (e.g., weekdays, holidays, long weekends), environmental factors (e.g., ambient temperature, relative humidity), and lag-based load patterns (e.g., previous day's load).

4) Model training utilizes K-Fold cross-validation, and performance is evaluated using standard forecasting metrics: RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error).

5) The study incorporates SHAP (Shapley Additive Explanations) to interpret model outputs, allowing for an understanding of feature importance and offering insights for practical improvements in energy management strategies.

1.4.2 Limitations

The limitations of this study are to consider three case studies as follows:

1) While the pure load is reconstructed by combining grid and PV data, minor discrepancies may exist due to unaccounted factors such as system losses or unmonitored sub-loads.

2) SHAP analysis for RNN-based models provides approximate feature importance and may not fully capture all intricate dependencies within deep neural networks.

1.5 Conception

The main idea of this research comes from the need for accurate and interpretable electricity load forecasting, which is essential for planning and managing energy systems in an era of rapid and uncertain energy usage patterns, especially in the context of complex power systems that require transparent, data-backed decision-making.

This research proposes a forecasting structure for electricity consumption using time-sequence deep learning models, including LSTM, GRU, Bi-LSTM, and Bi-GRU. Moving Average, DNN, CNN-LSTM, and Attention-LSTM are compared to obtain comprehensive results. Electricity consumption data is organized using Day-Aware Data Arrangement to clearly distinguish electricity usage patterns. Feature engineering is performed to identify influential variables, and model robustness is evaluated using twelve carefully designed test sets representing different times and operating conditions. To increase the reliability and understanding of the model, the system applies SHAP (Shapley Additive Explanations) to explain the model results and uses Mutual Information (MI) to select non-linear features, as well as Time Series Cross-Validation (TSCV) to evaluate the results comprehensively according to the context of time-sequence data, covering accuracy indicators of RMSE, MAE, and MAPE to evaluate the model's capabilities in multiple dimensions.

The integration of these components aims to create an electricity load forecasting system that is not only highly accurate but also whose results can still be interpreted well, are reliable, and can be confidently applied in real situations.

1.6 Research Benefits

This research presents a practical and interpretable deep learning-based framework for one-day-ahead electricity load forecasting, using real-world data from a university campus with integrated solar PV systems. By reconstructing the pure load and incorporating contextual features such as weather, time, and holidays, the model better reflects actual electricity demand and addresses forecasting challenges in renewable-integrated environments.

At the institutional level, the model supports more accurate energy planning, demand-side management, and cost optimization. At a broader level, it serves as a scalable solution for similar commercial or academic settings, promoting efficient energy use and integration of renewable energy. At the national level, the study aligns with the goals of energy transition and smart grid development by enabling more flexible, data-driven, and responsive electricity systems that can support peak demand management and sustainability targets.

By combining forecasting accuracy with explainability through SHAP, the research contributes not only to academic advancement but also to the foundation of smarter and more sustainable energy infrastructure.

1.7 Structure of thesis

This thesis is organized into five chapters, outlined as follows:

Chapter I: This chapter presents the background and significance of the thesis, including the challenges arising from complex variations in electrical load patterns. It discusses the problem, research objectives, scope and limitations, the proposed methodology concept, and the overall structure of the thesis.

Chapter II: This chapter covers theoretical principles and research on electrical load forecasting models, including a review of relevant optimization methods, feature analysis, and gap analysis in existing research.

Chapter III: This chapter presents one-day forecasting of electricity consumption using supervised deep learning techniques combined with daily data sorting to select models most likely to be further developed for maximum accuracy and to find the most efficient data management methods for achieving optimal results.

Chapter IV: This chapter discusses an advanced data-driven load forecasting framework for high-volume solar power systems: a case study of the SUT campus and data from ISO New England, demonstrating data preparation, model selection, simulation, and feature analysis using various tools for forecasting accuracy and reliability.

Chapter V: The final chapter summarizes the key research findings and evaluates the effectiveness of the proposed methodology, including a comparison of the results of the models and feature analysis tools from simulation and analysis results.

มหาวิทยาลัยเทคโนโลยีสุรนารี

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Load forecasting is the heart of power system management in the digital age, especially in the context of smart grids that require short-term and long-term forecasting accuracy to increase efficiency in planning energy production and distribution with stability and efficiency. Due to the complexity of time series data that is nonlinear and has various environmental factors that affect electricity loads, such as weather, consumer behavior, working days or holidays, etc., it is necessary to develop analysis techniques that can handle such data appropriately. In recent years, deep learning techniques have been increasingly popular for load forecasting, especially sequence models. However, the accuracy of forecasting models does not only depend on the model structure but also on appropriate feature selection and feature engineering that allow the model to learn data more deeply. In addition, model explainability is another important issue, especially in systems that require transparency and reliability to meet the requirements of accuracy, explainability, and appropriate feature selection. This research is interested in the approach that focuses on developing an explainable time series model by combining contextual feature selection and temporal cross-validation to obtain reliable results in the context of real load forecasting. The following topics will explore the knowledge related to various research areas, including data preparation and feature selection techniques, the development of deep learning time series models, model description techniques, and comprehensive model evaluation approaches in load forecasting.

2.2 Literature Overview

To provide an overview of past approaches to load forecasting using deep learning techniques, this research has compiled and compared related research, divided into 2 groups: the research group using deep learning models for load forecasting and evaluating model performance and the research group focusing on feature selection and creation and model explainability methods. Table 2.1 summarizes the main components of each research, including feature engineering, selecting the best model from forecasts compared to other models or categories, metrics used for evaluation, techniques used for feature selection, and model explainability.

From the analysis of the research in Table 2.1, many works using various deep learning models in the context of load forecasting can teach temporal relationship and non-linear behavior better, along with the selection and design of specific features in contextual feature engineering. However, there is still a lack of non-linear feature selection, evaluation by time-based data partitioning, and the clear description of the model results. Therefore, our research aims to develop an approach that combines the selection of non-linear features using MI, time-based cross-validation for time series data, and model description using SHAP for deep analysis under the framework of LSTM, GRU, Bi-LSTM, and Bi-GRU models to create an accurate and easy-to-understand load forecasting framework.

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.1 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting

Ref	Feature Engineering	Best Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Panapongpakorn & Banjerdpongchai, 2019	✓	LSTM	✓	✓	-	MI	-
Vanitha & Swaroopan, 2023	✓	Bi-LSTM	✓			Correlation	-
Mpawenimana et al., 2020	STDAN	LSTM	-	✓	✓	Active Power	-
Prakash et al., 2023	✓	LSTM, Bi-LSTM	✓	✓	✓	-	-
Habbak et al., 2023	✓	NN	✓	-	✓	-	-
Atef & Eltawil, 2020	✓	Bi-LSTM	✓	✓	✓	-	-
Pavlatos et al., 2023	✓	Bi-LSTM	✓	-	✓	-	-

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.1 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting (Continued)

Ref	Feature Engineering	Best Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Sias et al., 2024	✓	Multi Bi-GRU	-	✓	✓	-	-
Abumohsen et al., 2023	✓	GRU	✓	✓	✓	-	-
Cordeiro-Costas et al., 2023	✓	LSTM	Normalized	✓	-	-	-

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.2 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research

Ref	Feature Engineering	Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Lee et al., 2023	✓	XG Boost	-	-	✓	-	SHAP value based
Neubauer et al., 2025	✓	ED, XG Boost	Normalized	✓	-	-	SHAP, Attention
van Zyl et al., 2023	✓	CNN	✓	-	✓	MI	SHAP, Grad-CAM
Jang et al., 2023	✓	RAID	-	-	✓	-	SHAP
Koprinska et al., 2015	✓	LR, BPNN	✓	-	-	MI, Relief	-
Pirbazari et al., 2019	✓	GBRT	-	✓	-	MI, RFE, Elastic Net, F-regression	-
Liu et al., 2024	✓	KAN, Transformer, Informer	✓	-	-	Correlation Feature Self-Assignment Module	KAN

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.2 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research (Continued)

Ref	Feature Engineering	Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Maarif et al., 2023	✓	SL-LSTM, DL-LSTM, Bi-LSTM	✓	-	-	SHAP	SHAP
Wang et al., 2025	✓	SMLR, XG Boost, LASSO-XG Boost	✓	✓	✓	SHAP	SHAP
Van Zyl et al., 2024	✓	1D-CNN	✓	✓	✓	SHAP, XG Boost, Feature Importance, Variance Threshold, NSGA-II	SHAP, Grad-CAM
Wu et al., 2020	✓	Attention-LSTM	✓	✓	-	Attention weights	Attention weights
Ullah et al., 2024	✓	CNN-LSTM	✓	✓	✓	Correlation	-
Fan and Hyndman, 2012	✓	Semi-parametric additive	✓	✓	-	-	-
Shi et al., 2018	✓	Deep RNN	✓	✓	-	-	-

มหาวิทยาลัยเทคโนโลยีสุรนารี

From Table 2.1, additional research related to the introduction of feature selection techniques and model description capability in load forecasting is summarized in Table 2.2, which summarizes the main components of each research study, including feature engineering, all models used in forecasting, evaluation metrics, feature selection techniques, and model description capability. The trend of developing load forecasting models focuses on interpretability and accurate feature selection to increase both efficiency and transparency. Many research studies focus on using SHAP and other XAI techniques to analyze feature importance. However, there are still gaps in using filter-based methods such as MI and temporal cross-validation, including using SHAP for feature selection, which our research will supplement to obtain a system that is both accurate, interpretable, and reliable in the long run.

2.3 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting

To delve deeper into the trends, technologies, and limitations of the related research in Table 2.1, this section describes in-depth details of each selected research, focusing on feature selection methods, techniques used for load forecasting, proposed models, evaluation metrics, and limitations of each work, and proposes ways in which we can improve upon those limitations within the framework of load forecasting with deep learning.

(Panapongpakorn & Banjerdpongchai, 2019) This research presents a non-linear feature selection (Mutual Information—MI) method, where the information relationship between two variables captures the non-linear relationship to select the important input variables for the model. This research aims to apply autoregressive integrated moving average (ARIMA) and seasonal autoregressive integrated moving average (SARIMA), artificial neural network (ANN), and long-short term memory (LSTM)-based recurrent neural network (RNN) to improve the performance of short-term electrical load forecasting (STLF) in energy management systems (EMS). Time series features were structured, and the performance was measured with Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The models from neural networks have

better performance than the models from time series analysis, with the best performing model being LSTM. However, the performance does not cover the division of train/validation sets by time, and there is no time-aware dataset division (temporal cross-validation), so the model still cannot learn well in all time periods. The technique used to describe the model, which features affect the model output, has not been mentioned, and there is still a lack of in-depth analysis of the specific model directly.

(Vanitha & Swaroopan, 2023) This paper proposes a method to compare Stacked GRU-RNN and Bidirectional LSTM (Bi-LSTM) artificial neural network models in forecasting electricity load and renewable energy production in Smart Grid systems using real energy consumption data along with the analysis of the relationship of the input data to select the most influential features from the use of correlation analysis. The feature selection is linear relationship statistics. The linear relationship between 2 variables has a point that it does not support non-linear relationships from some features, so it does not cover all the impacts of the features on the model output. The results show that the Bi-LSTM model provides high accuracy in energy and load forecasting.

(Mpawenimana et al., 2020) This paper compares the short-term and medium-term energy load forecasting capabilities of LSTM and ARIMA models in the context of a smart home with an energy harvesting system. A new data preparation technique called STDAN (Same Time a Day Ago or Next) is proposed for handling missing values, which performs better than the average or previous value filling method. The paper states that the prediction uses only active power data from the dataset, which can be considered a feature selection, focusing on a single feature believed to be related to the energy load, which is a manual selection. The experiments show that LSTM significantly outperforms ARIMA in all cases. More sophisticated feature engineering can also be developed to help the model learn better. There is no mention of efficient feature selection and no explanation of which features influence the results.

(Prakash et al., 2023) This paper presents a forecast of the daily COVID-19 cases in India over the next 6 months using time series analysis techniques such as machine learning and deep learning, as well as a hybrid model. To compare the performance of different models in predicting the number of infected people using historical data

and evaluating the performance through statistical indicators. The experimental results show that basic models such as Prophet, Vanilla LSTM, and Bi-LSTM give more accurate results than complex or hybrid models. This research is not directly in the context of energy or electrical load, and there is no feature selection or model description.

(Habbak et al., 2023) This article is a systematic review covering load forecasting techniques in smart grids. The analysis results show that techniques using Machine Learning (ML) and Neural Networks (NN) provide the highest forecasting accuracy, with significantly lower Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) values than other methods. No new methods or in-depth comparisons were found, efficient feature selection, and lack of approaches to explain causal models.

(Atef & Eltawil, 2020) This article presents a comparative study between unidirectional (uni-LSTM) and bidirectional (bi-LSTM) LSTM models in 1, 2, and 3-layer nesting, using an automatic parameter tuning tool to find the best model. The experimental results show that Bi-LSTM significantly outperforms Uni-LSTM and Support Vector Regression (SVR) in terms of forecasting accuracy. When considering error values such as RMSE, MAPE, and MAE, deep nested models do not significantly outperform single-layer models, and they take longer to compute. There is no mention of efficient feature selection, and there is no explanation of how each feature affects the other.

(Pavlatos et al., 2023) This paper proposes a Bi-LSTM based neural network-based electrical load forecasting framework that can simultaneously process historical and future data to improve the forecast accuracy of temporal data. Experiments on multiple datasets show that Bi-LSTM significantly outperforms RNN, LSTM, and GRU. MAE and RMSE are used for performance evaluation. It is found that additional performance evaluation techniques can be added to cover more comprehensive models. There is still no efficient feature selection, and there is a lack of approaches to explain the causal model.

(Sias et al., 2024) This research presents a multivariate bidirectional GRU model combined with periodic features for forecasting energy loads and renewable energy sources (e.g., solar cells). The model uses multivariate data, such as system energy

data and weather data (for power generation), and tests are conducted on daily, hourly, and minute data levels. The experimental results show that the proposed model has the highest accuracy. And it takes less processing time than the basic Bi-GRU and Bi-LSTM models. From this research, it is not clear which features affect the model performance, and it lacks the step of explaining the relationship of the features to the results.

(Abumohsen et al., 2023) This paper uses deep learning techniques to forecast electrical power loads: Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNN). By testing all three models, it was found that the GRU model gave the best results in terms of accuracy and the lowest error. There are still limitations found in this research: the lack of feature selection steps and model result explanations, and the analysis only analyzes the results, not the components of the model. It is possible to analyze the behavior of the GRU in more detail.

(Cordeiro-Costas et al., 2023) This paper focuses on evaluating and comparing the accuracy of various artificial intelligence (AI) techniques in forecasting electrical loads in buildings. To compare with the forecasting of loads in single-family homes in the United States, using a 10-fold cross-validation technique to assess the bias and variance of each model. The experimental results found that LSTM is the model with the lowest error distribution in the validation and test datasets. Using cross-validation General models may not be suitable for time series data and lack in-depth analysis of feature selection.

From the research studies in the past few years, there is a clear trend in applying deep learning techniques together with the selection and design of specific features, or so-called contextual feature engineering, to address the complexity of electrical load data in real contexts. Traditional models such as ARIMA or linear regression often cannot effectively handle behaviors that change with context, such as days of the week, weather conditions, or holidays. While deep models such as LSTM and GRU can learn temporal relationships and non-linear behaviors better. However, these models are most effective only when they receive input data with appropriately designed features, such as adding calendar effects, temperature, humidity, sunlight, and various time indicators such as rush hours or public holidays, which allow the

model to learn complex patterns of electrical loads more accurately. Therefore, research in this direction is continuously increasing, focusing on both developing model architectures and enhancing the potential of input data.

2.4 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research

To address the challenge of developing a load forecasting model that is not only accurate but also transparent and interpretable, several recent studies have focused on appropriate feature selection, comprehensive evaluation, and the use of Explainable AI (XAI) techniques to enhance the reliability of the model. This chapter presents an in-depth analysis of those studies, examining the key elements of the model, feature selection, evaluation, and model explanation, and points out the limitations and solutions that will be further developed in our work. From Table 2.2, it can be explained as follows:

(Lee et al., 2023) Using Explainable AI for Transparency in Load Forecasting, we propose a new metric that makes the feature importance analysis more reliable, supporting applications in real systems that require transparency, such as energy systems. Since the work has not been tested with multiple datasets or models, which may limit the generality of the metric.

(Neubauer et al., 2025) This paper presents a multi-step heating load forecasting model that emphasizes the ability to explain AI using SHAP values and temporal attention mechanisms to enhance the understanding of the process of data-driven forecasting models. Comparison with other explainable techniques such as MI is not discussed.

(van Zyl et al., 2023) This article analyzes and compares the capabilities of two types of Explainable Artificial Intelligence (XAI) techniques: Grad-CAM (Gradient-weighted Class Activation Mapping) and SHAP (Shapley Additive Explanations) in performing feature selection for energy demand forecasting models from large time series using a multi-headed CNN as the model base. CNN models may not be suitable

for time series load data with clear temporal dependency and lack of comparison with other models for comprehensive results.

(Jang et al., 2023) This article presents a Robust and Interpretable DPLF (RAID) model for Daily Peak Load Forecasting (DPLF) that is both accurate and interpretable using multiple Deep Neural Networks (DNNs) with SHAP values analysis to explain the importance of features. However, there is no evaluation of temporal generalization or cross-time validation.

(Koprinska et al., 2015) This paper presents a two-step feature selection method for 5-minute electricity load forecasting. Initially, features are selected based on daily and weekly electricity usage patterns, such as load from the previous hour, and feature selection techniques such as Relief and Mutual Information are used to assess the importance of features and select the most suitable feature set. The results show that using this feature selection method can reduce the number of features without compromising the forecast accuracy. However, the work has not included the interpretability perspective from XAI or analyzed the impact of deep features.

(Pirbazari et al., 2019) studied the effectiveness of feature selection methods in short-term household energy forecasting, especially in the context of smart meter data. It shows that feature selection techniques can effectively screen out suitable variables specific to each household, speed up model training, reduce model complexity, and produce forecasting results that are close to or better than those without feature selection. This research lacks the mention of deep model description techniques.

(Liu et al., 2024) This paper proposes a new model called Correlation Feature Self- Assigned to flexibly combine features based on the degree of data correlation, the Interpretable Kolmogorov-Arnold Network (KAN) is the ultimate predictor for forecasting various types of energy loads in IES. There is no comparison with other XAI techniques such as SHAP or MI, which are suitable for non-linear data.

(Maarif et al., 2023) This research successfully developed an energy forecasting model using LSTM combined with XAI, which is not only efficient in forecasting but also clearly explains the model results, allowing users to understand and trust the results obtained from the model. The Bi-LSTM and double-layer LSTM models perform

slightly better than single-layer LSTM, with lower RMSE. This research uses SHAP to select important features and train a new model, which significantly improves forecasting performance. However, there is no mention of feature selection before model training. Immediately introducing all features into the model may cause overfitting or noise from unimportant features. The work does not mention the use of techniques such as k-fold cross-validation or other data partitioning to prevent overfitting. A single train/test partition may result in model bias or incorrect performance evaluation in real-world use cases.

(Wang et al., 2025) This research demonstrates the effectiveness of using the SHAP technique to analyze the importance of features in a short-term load forecasting model by identifying the features that have the most impact on the model results, which helps to improve the model accuracy and reduce the complexity of the model by removing unnecessary features. Using the SHAP technique also increases the transparency of the model operation, allowing users to understand and verify the model operation effectively, which is very useful for applying the model in real situations that require reliability and transparency. The research found that the model description from SHAP has not been linked to a systematic feature selection strategy.

(Van Zyl et al., 2024) This article presents the ability of SHAP to reduce the number of features without significantly degrading the model performance. In some cases, it helps to make the model more stable and improve the interpretation of the results. However, it is not suitable as a method for selecting “initial” features but is suitable for fine-tuning after obtaining the initial feature set. Although SHAP is not a direct feature selection technique, the researchers use SHAP scores to rank the features and then build a new model by comparing it with other feature selection methods. Therefore, SHAP is used strategically to “select features by importance.” Although this work uses a variety of feature selection methods, it still lacks a filter method that supports non-linear relationships. Independent of the model, such as MI, the addition of MI enhances the "pure data" view that is independent of the structure or bias of any model. In addition, a single model (1D-CNN) is used as the basis for analysis. The results of SHAP depend on the model used. If other models are used for comparison, such as LSTM or Bi-LSTM, the ranking of features may be different. Using only one model may

limit the reliability of the conclusion of "which features are important" after model training. And cross-validation is used to evaluate the accuracy of the model for each selected feature set.

(Wu et al., 2020) This paper proposes a combined deep learning framework based on an attention-based LSTM model to improve short-term forecasting performance by selectively focusing on important temporal features. The attention mechanism enhances the model's ability to capture key dependencies in time-series data and improves prediction accuracy. However, the study does not perform explicit feature selection, and the explainability of the model is limited to attention weights without providing a comprehensive feature-level interpretation such as SHAP.

(Ullah et al., 2024) This paper presents a comprehensive review and simulation study of short-term load forecasting using CNN–LSTM hybrid models, highlighting their ability to extract spatial features via CNN and temporal dependencies via LSTM. The results demonstrate improved forecasting accuracy compared to single models. However, the study mainly focuses on performance evaluation and lacks feature selection analysis and model explainability techniques to interpret how input features influence the forecasting outcomes.

(Fan & Hyndman, 2012) This paper proposes a semi-parametric additive model for short-term electrical load forecasting by decomposing load patterns into trend, seasonal, and smoothing components using statistical feature engineering techniques. The model provides good interpretability and stable forecasting performance; however, it is limited in capturing complex nonlinear relationships and does not incorporate feature selection or advanced model explainability methods.

(Shi et al., 2018) This paper proposes a deep learning-based load forecasting framework using a novel pooling deep recurrent neural network to capture complex temporal dependencies in household electricity consumption. The proposed model significantly improves forecasting accuracy compared to traditional methods; however, it treats the model as a black box and does not provide explicit feature importance analysis, feature selection, or explainability to support model interpretability and reliability.

While deep learning models have become increasingly popular due to their ability to produce highly accurate results, a major problem is the interpretability of the models, which often have a black box appearance. That is, even if the results are accurate, they cannot clearly explain which factors the model uses to make decisions. In the context of energy management in institutions or organizations, accurate forecasting alone may not be enough if decision-makers cannot understand or trust the decision-making process of the model. Therefore, modern research has begun to focus on interpreting model results using techniques such as Shapley Additive Explanations (SHAP), which can indicate how much each feature influences the model's results. This technique allows for a transparent analysis of which factors drive the forecast results, such as temperature, time of day, or day of the week, and can be used to further develop energy policies or improve models in the future. Combining the ability to explain results with forecast accuracy is an important direction in the development of AI models for sustainable energy management.

2.5 Research Gaps in Load Forecasting Framework

From the review of past research, it was found that there is a tendency to use various deep learning models, such as LSTM, Bi-LSTM, and CNN, in the context of energy load forecasting, both short-term and long-term, which allows the model to learn the temporal characteristics of the data efficiently. However, many researchers still lack feature selection in a way that supports non-linear relationships, including model evaluation in a way that is appropriate for temporal data, such as time-aware data partitioning or temporal cross-validation. In addition, although many researchers have started to use Explainable AI (XAI) techniques such as SHAP and Grad-CAM to analyze the importance of features and increase the transparency of the model results, there are still significant gaps in the application together with filter-based feature selection methods such as Mutual Information (MI), which can help filter important features directly from the data, independent of the model, and reduce bias from the model structure itself. In evaluating the forecast results, most researchers use various indicators to measure the performance of the model in many dimensions, including the accuracy of the prediction value, the average absolute error, and the relative error

to the actual value. Using these indicators together allows for a more comprehensive analysis of the quality of the model, especially when applied in real situations that require both accuracy and reliability.

Based on the above findings, this research aims to develop a comprehensive load forecasting approach encompassing three main dimensions: feature selection using mutual information to support nonlinear relationships; modeling results through Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models for multi-dimensional model coverage; detailed model description within the NN group using SHAP; evaluation using Temporal Cross-validation suitable for timeline data; feature development and enhancement using Feature Engineering to suit data characteristics; and the use of multi-dimensional metrics including RMSE, MAE, and MAPE. This will result in a developed forecasting system that is accurate, transparent, and interpretable, suitable for use in smart energy systems requiring long-term reliability.

มหาวิทยาลัยเทคโนโลยีสุรนารี

CHAPTER III

Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement

3.1 Introduction

In the context of electrical load forecasting, choosing the right method for the highly complex data is an important issue, especially for electrical load data that has a non-linear relationship and changes significantly over time, such as weekdays, holidays, or school openings and closings, which all affect energy use behavior at the institutional or large organization level, such as universities.

Current research trends have begun to focus on the use of deep learning models, especially the RNN family of models such as LSTM, GRU, Bi-LSTM, and Bi-GRU, which are designed to efficiently handle time series data. The strength of these models is their ability to learn patterns of past data to predict the future. However, these models often have limitations in terms of transparency in explaining their results, which raises questions about their practical reliability. To solve this problem, research focuses on the process of selecting appropriate features, adding contextual information such as time, weather, calendar, and holidays, and using model testing and evaluation techniques that are consistent with time series data, such as data partitioning by the Expanding Window method, which does not leak future data to model training. In addition, the use of the SHAP technique allows for a clearer explanation of model decisions, both at the individual data point level and in general.

This chapter aims to describe the research process used to build and test the electrical load forecasting model, which will serve as an important basis for the comparative performance analysis in the next chapter. Parts of this thesis have been published in the proceedings of the 2025 13th International Electrical Engineering Congress (iEECON2025), is shown in the Appendix D.

3.2 Data Collection and Preparation

The datasets collected and used in the forecasting include electric load data from utility meters, which shows historical data from the Automatic Meter Reading (AMR) system maintained by the Provincial Electricity Authority (PEA), which records electricity usage according to load demand at the university, as well as solar energy production data, which records energy production from meters installed in the area with solar panels at Suranaree University of Technology in Thailand. Data regarding compensatory holiday long weekends is gathered from university announcements and the academic calendar. A reliable public website with an API for direct data extraction is the source of historical condition data. Every 15 minutes for a year, from January to December 2023, these statistics are collected and compiled. The electrical load forecasting utilizes 35,040 datasets in total. In the data cleaning process, which is an important process in preparing various data before using it for forecasting in the model, the following methods are used: Handling missing values (Handling Missing Values) will delete the missing data range or the data range with abnormalities and add new data from the method of calculating the average and transforming the data format to be appropriate (Data Formatting) using Timestamp to help create the Time Factor and Social Factors features, helping the model to learn the relationship of the data better and be more efficient in forecasting. Since the electrical load in the university is a high PV penetration load with a proportion of solar energy produced and sent to the total power system, the electric load data from utility meters or net load obtained is data that is mixed with solar energy production data or PV generation, making it possible to see electricity usage according to the load demand that is unrealistically low. Therefore, PV Generation is added back to obtain pure load or electrical load data that can be used to assess the actual electricity demand in forecasting.

The data is divided by taking the entire cleaned dataset for the year and splitting it into three sets including the training set, which uses 80% of the total data; the validation set, which uses 20% of the data for parameter tuning; and the test set, which consists of a separate portion of data that the model has never seen before, reserved for final evaluation.

3.3 Feature Extraction

Assigning features to existing data is very important in forecasting electrical load. Because features are variables that will be fed into the model to help the model learn relationships and generate accurate forecast results. Features used in forecasting this type of data are a collection of parameters that are significant to load forecasting and are created by considering the context of time and day. Creating features from data can be divided into two types as follows: The first type is creating features that are Day-Aware Data Arrangement, which is a method of arranging data that takes into account the context of time and day to allow the model to learn the characteristics of the data on each day or specific time period. It consists of Hour of Day, which is a value that specifies the hours in one year; Quarter-Hourly Interval, which is a value that specifies the minutes every 15 minutes in each hour; Weekdays and Weekend, which are values used to separate data between weekdays and weekends; Daytime and Nighttime, which are values used to separate data between day and night; Compensatory Holiday, which is a value that specifies whether that day is a holiday or not, Long Weekend, which is a value that specifies that period is a special holiday or festival. The second type is a feature that is weather data, consisting of ambient temperature data and relative humidity data. The number of data sets used to create and divide the feature data sets will be the same as the data used in forecasting electrical load.

3.4 Feature Engineering

Feature engineering is an important step in developing a load prediction model, playing a role in creating appropriate variables that accurately represent temporal behavior, which will improve both the accuracy and explainability of the model. From the collection and review of research, feature engineering techniques will be used. In this research, the techniques will be divided into two main groups: feature generation and external features. The first group is featuring generation, which is the creation of new features from existing temporal data and the transformation of temporal data that may not be easily understood by the model into logical information that is meaningful according to the context. as described as follows.

3.4.1 Feature Transformation / Binarization

Variable transformation and binarization are the processes of transforming the source data into a format that is easy to interpret or respond to the model, as follows: Create a feature that divides Monday to Friday to represent school days and Saturday and Sunday to represent weekends, with the numbers 0 and 1, so that the model clearly understands the difference between Monday and Sunday. Create a feature that divides day and nighttime with the numbers 0 and 1, such as 8:00 - 17:00, which is daytime and is 1, and the rest is nighttime, which is 0. Dividing day and nighttime into categories that are easier to understand. Creating a feature that does not just divide normal holidays and school days but can be adjusted according to special events, such as compensatory holidays or school days, helps the model better understand the behavior at each time period.

3.4.2 Time-based Feature Engineering / Ordinal Time Block

Segmenting data into meaningful timelines with 1 year of data every 15 minutes creates a feature that allows the model to learn the day and time, such as day 1, where 00:00 - 00:45 will be set as 0, time 01:00 - 01:45 will be set as 1, time 02:00 - 02:45 will be set as 2, and so on until 23:00 - 23:45 will be set as 23, considered day 1, and day 2 is set as the same. The method and advantages of this feature are as follows:

Ordinal Encoding: Use sequential numbers (0, 1, 2, ... 23) to represent time of day, which helps the model to recognize the continuity of the data.

Time Segmentation: Divide time periods into groups with different usage patterns, such as morning (Morning), afternoon (Afternoon), evening (Evening), and night (Night).

Pattern Recognition: The model can learn the trends of each time period, such as energy consumption may be higher in the morning or lower at night.

Use this feature to forecast energy usage by letting the model understand that different days can have different energy usage patterns, helping the model better understand the context of that time period.

3.4.3 Time Component Extraction

Decomposing time components into separate features creates a new feature that describes each sub-hour: 00.00 is 0, 00.15 is 15, 00.30 is 30, and 00.45 is 45. After 01.00, it is 0; 01.15 is 15; 01.30 is 30; and 01.45 is 45. After 02.00, it is 0. This continues so that the model can capture the relationship between time periods and different load behaviors. The model can understand the recurring behaviors during each sub-hour, and this helps to understand the trends in energy usage over time. For example, every hour from 0-45 may have a similar energy usage pattern.

3.4.4 Time Series Features / Lag Features

Building features from previous values (lag) is an important basic technique in the time series data analysis. It uses historical load values to build features such as load in the same period of the previous day (Load t-1d). This method allows the model to learn trends and seasonality efficiently.

In the second group there are external (exogenous) features, external variables that do not arise from the process of the target variable itself but have an impact or relationship with the forecast of the target variable and can be used as additional input to the forecasting model to increase its accuracy, consisting of the following techniques:

3.4.5 Ambient Temperature

Ambient temperature is an external factor that directly affects energy use behavior, such as air conditioning or heating use. This information is often used in conjunction with load data to help models understand the physical relationship between weather conditions and energy demand.

3.4.6 Dew Point

The Dew point is a measure of humidity in the air, which affects temperature perception and the use of cooling or heating equipment, especially in regions with high humidity. Incorporating dew point values into models allows for a more accurate capture of energy consumption behavior.

3.5 Advanced RNN Architectures

The design of the model architecture is the heart of developing the electrical load forecasting model. Due to the nature of electrical load data being a time series with significant variations over time and period, selecting a model that can accurately capture the characteristics of the data sequence is essential. In this section, we have selected the architecture of the Recurrent Neural Networks (RNNs) model that has been improved to better remember long-term data, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), as well as the Bidirectional version (Bi-LSTM and Bi-GRU), which can see the context from both the past and the future in the time sequence, allowing a deeper understanding of the relationships in the data.

3.5.1 LSTM Architecture

The Long Short-Term Memory (LSTM) model is a structure of Recurrent Neural Network (RNN) designed to solve problems and manipulate time-series data. It consists of a Forget Gate that determines which data should be forgotten from the original cell state. Let x_t be the input feature vector at time t, $\sigma(\cdot)$ be the sigmoid activation function, h_{t-1} be the hidden state from time $t-1$, and b_f be the bias vector of the forget gate, W_f be the learnable weight matrix of the forget gate. The forget-gate output is computed as shown in Equations (3.1).

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \tag{3.1}$$

where $f_t \in [0,1]$ is close to 1, indicating retention, while values close to 0 indicate erasure. This helps determine whether data should be retained. The next component, the Input Gate (i_t) and the Candidate Cell State, decides which new data to add to the cell state, as shown in Equations (3.2) and Equations (3.3).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\tilde{C}_t = \sigma(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.3)$$

where i_t controls which new data should be imported, and $\tilde{C}_t$ represents the new data processed (candidate memory). The next component, the Update Cell State, combines the Forget Gate and the Input Gate to create a new cell state, as shown in Equations (3.4).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.4)$$

where the forgotten data is multiplied by f_t, and the new, important data is multiplied by i_t and added together. The final section is the Output Gate (o_t) and Hidden State. These define the data to output as hidden states for future forecasting, as shown in Equations (3.5) and Equations (3.6).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.5)$$

$$h_t = o_t * tanh(C_t) \quad (3.6)$$

where h_t is the hidden state, used to predict the electrical load at the next point.

LSTM is a special kind of RNN that can identify long-term dependencies. LSTMs are purposefully designed to avoid the problem of long-term reliance. They essentially behave in such a way that their default state is long-term memory retention. For time series forecasting, they are hence the perfect parallel. Fig. 3.1 describes how LSTM cells are represented graphically.

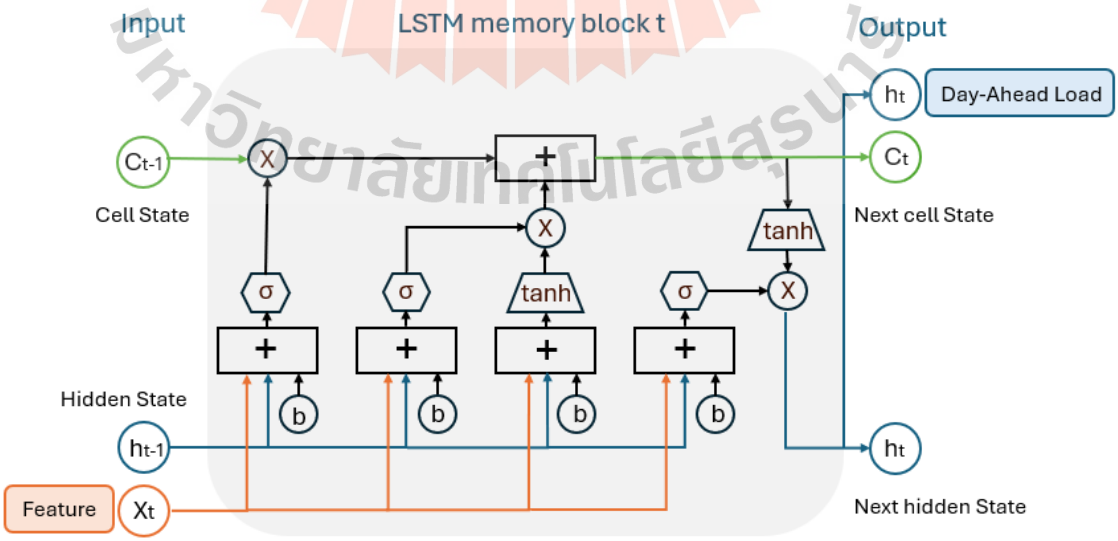


Figure 3.1 LSTM architecture

3.5.2 Bi-LSTM Architecture

The Bidirectional Long Short-Term Memory (Bi-LSTM) model is a structure that is developed from LSTM to increase the ability to learn sequences of data that have relationships in both the past (past dependencies) and the future (future dependencies) simultaneously. It consists of two LSTM layers that work together: a forward LSTM that processes data from the past to the present and a backward LSTM that processes data from the future back to the present at any time t. The results of both directions are combined into a single hidden state as shown in Equations (3.7).

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \tag{3.7}$$

where $\vec{h}_t$ is the hidden state from the Forward LSTM, $\overleftarrow{h}_t$ is the hidden state from the Backward LSTM, and h_t is the combination of the two results.

Bi-LSTM is a sequence processing model that is made up of two LSTMs, one of which accepts input in a forward direction and the other in a backward manner. Bi-LSTMs give the network access to additional information, which improves the context of the algorithm. The primary advantage is that each input sequence piece includes both current and historical data. Therefore, Bi-LSTM can produce a more pertinent outcome by merging LSTM layers from both directions. The configuration of LSTM layers in Bi-LSTM is shown in Fig. 3.2.

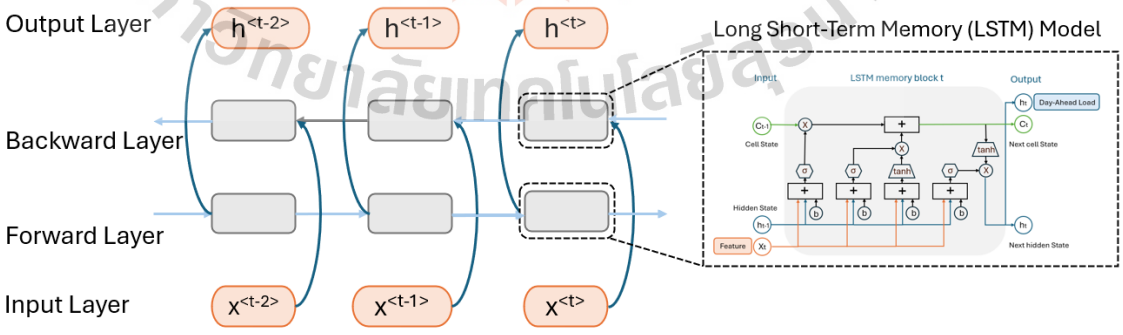


Figure 3.2 Bi-LSTM architecture

3.5.3 GRU Architecture

GRU resembles LSTM, which regulates data flow using gates. But with a simpler design, GRU provides an improvement over LSTM. GRU just has a hidden shape, not a distinct cell state like LSTM. The GRU has unrestricted access to all concealed states. Compared to LSTM units, GRUs are easier to train and do not require a memory unit to regulate the data flow. As illustrated in Fig. 3.3, GRU may train more quickly or employ less data to generalize since it has fewer parameters.

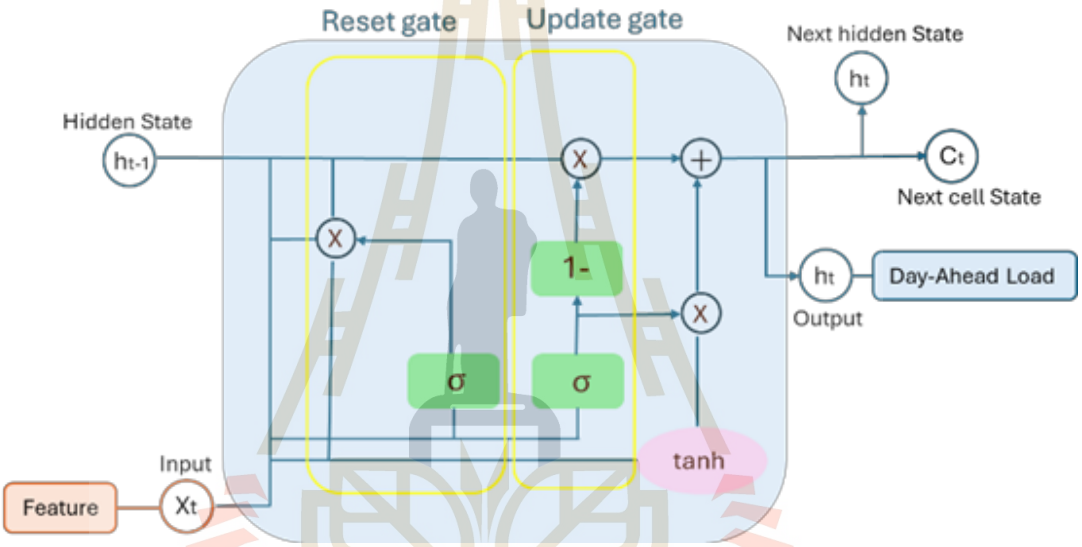


Figure 3.3 GRU architecture

3.5.4 Bi-GRU Architecture

Bidirectional GRU - A bidirectional GRU, often known as a Bi-GRU, a bidirectional GRU is designed to analyze univariate time series data. This enables the temporal dependencies found in the electrical load data to be accommodated. This integration becomes essential for precise forecasting by utilizing data from both previous and subsequent time periods. As illustrated in Fig. 3.4, this functionality plays a substantial role in improving forecasting precision.

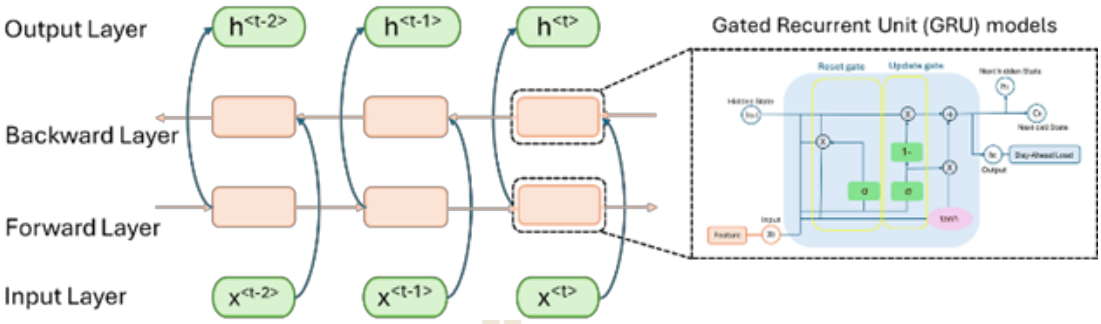


Figure 3.4 Bi-GRU architecture.

3.6 Workflow of the Proposed Method

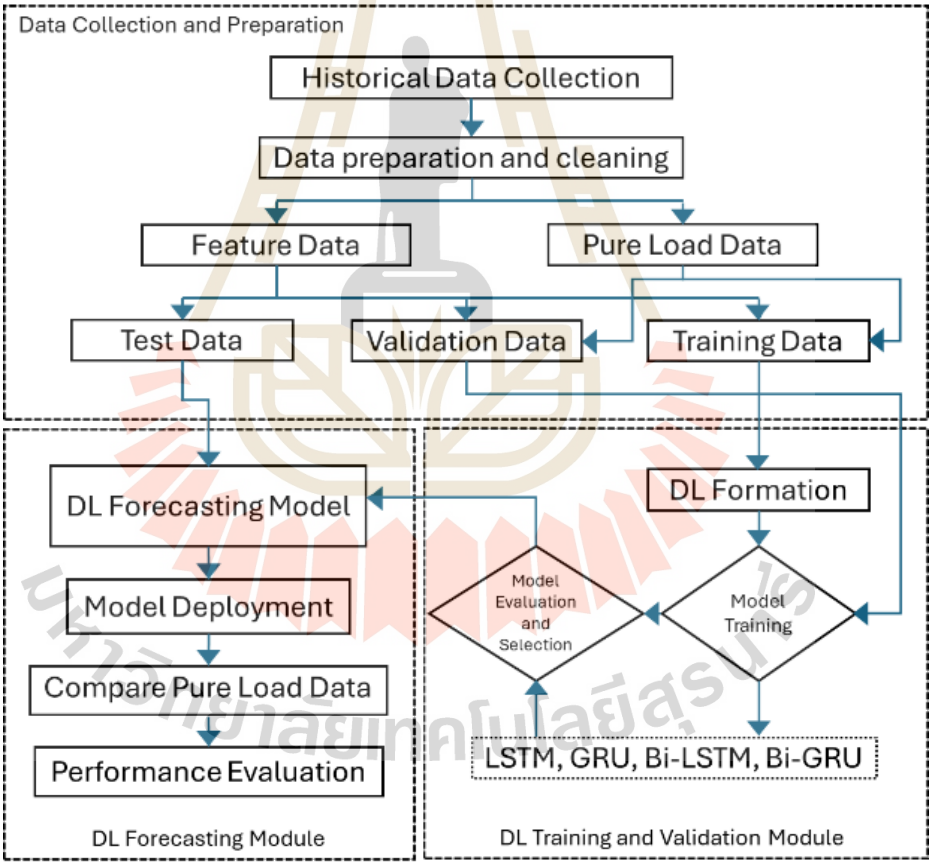


Figure 3.5 The Proposed Load Forecasting Workflow

The electrical load forecasting, including finding the model with the appropriate accuracy for the university's load data, can be done as follows. Start by collecting all relevant historical data, then clean the data with necessary techniques to facilitate

model learning. The data is divided into three groups: the training data group receives feature data as input and pure load data for the model to learn the relationship, then performs the deep learning formation step, which is to design the structure and select the function for the various models and train the model. The models used for training include LSTM, GRU, Bi-LSTM, and Bi-GRU, and they output the load forecast. The validation data group receives the same input as the first group, receiving various models for forecasting. When the output is the load forecast, it will help check the accuracy of model training and prevent overfitting in the model training process. It also helps evaluate the performance of the model during model training of the predicted values and actual load values. Then select the best-performing model as the deep learning forecasting model. The test data group receives features as input, inputs data into the selected model, and outputs daily load forecasts and obtains the developed model. By comparing the predicted values with the actual load values available, the performance of the tested model can be evaluated, as can be seen in Fig. 3.5.

3.7 Performance Factor

To examine the performance of the proposed forecasting model through evaluation metrics including percentage of MAPE, MAE and RMSE, as shown in Table 3.1. Let Y_i denote the actual load at time i, and $\hat{Y}_i$ denote the corresponding predicted load.

Table 3.1 Performance Factor

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y}_i)}{Y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert}{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.8 Simulation Results

This section presents experimental results and performance analysis of the electrical load forecasting models developed in Chapter 3. It compares the forecasting results of LSTM, GRU, Bi-LSTM, and Bi-GRU models on pre-prepared test datasets, using RMSE, MAE, and MAPE performance indicators to measure model accuracy across multiple dimensions. The analysis is divided into three main parts: evaluating model learning and accuracy during the validation and training phases and comparing forecasting results between weekdays and weekends. This chapter aims to answer the key question of which model is best suited for electrical load forecasting in the context of real-world data in terms of accuracy, consistency, and operational capability under various scenarios, and to evaluate the performance of the best model on different datasets in terms of time and context.

3.8.1 Experimental setup

Python 3 software was employed to simulate the Suranaree University of Technology's daily electrical load forecast in order to create a model suitable for the university's electrical loads and evaluate and contrast efficiency using evaluation metrics including MAPE, MAE, and RMSE. The data selected for use was every 15 minutes throughout 2023 and contained, in addition to pure load and feature or electric load data used for forecasting, The models made use of to simulate the results were LSTM, GRU, Bi-LSTM, and Bi-GRU. To get this data ready for entry into the model, it was separated into three sections. This section will show the outcomes of the simulation and different model settings.

In preparing data for power load forecasting, this research addresses missing data using filler methods that correspond to the temporal structure of the data. However, data filler can affect model performance, potentially reducing power load fluctuations and causing bias in the learning process, especially when the filled data is large or concentrated over the same time period. Furthermore, filler values can impact the generation of temporal features, such as lag features, which rely on data continuity. To limit these effects, this research controls the proportion of filled data to

a low level and selects filler methods appropriate to the data characteristics. However, future research should develop more advanced methods for handling missing data or perform model sensitivity analysis to filler to improve forecasting reliability.

The selection of hyperparameters was carried out using a trial-and-error method within a systematic evaluation framework. It began by referencing commonly used values from relevant research, then adjusting parameters individually or in sets to study their impact on the model's learning ability and forecasting errors. This method allowed for observation of the model's behavior under different settings and clearly identified trends in values affecting model performance. The final hyperparameter selected is the set of values that yields the best accuracy results, consisting of the following: A small data set of size four is used to improve the diversity of learning, using 50 epochs to give the model enough time to learn from the training data, and the model structure is set to have 100 units as the number of units in the hidden layer to train different DL models. Two layers of each model are used, a dropout rate of 0.2 is used to prevent overfitting, and Adam Optimizer is used to adjust the model weights, and a learning rate of 0.001 is used as the learning rate for Adam Optimizer, as shown in Table 3.2.

Table 3.2 Model parameter settings

Methods	LSTM	GRU	Bi-LSTM	Bi-GRU
Layer	2	2	2	2
Units	100	100	100	100
Dropout rate	0.2	0.2	0.2	0.2
Learning rate	0.001	0.001	0.001	0.001
Batch Size	4	4	4	4
Epochs	50	50	50	50
Optimizer	Adam	Adam	Adam	Adam

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.8.2 Model performance in validation and model training

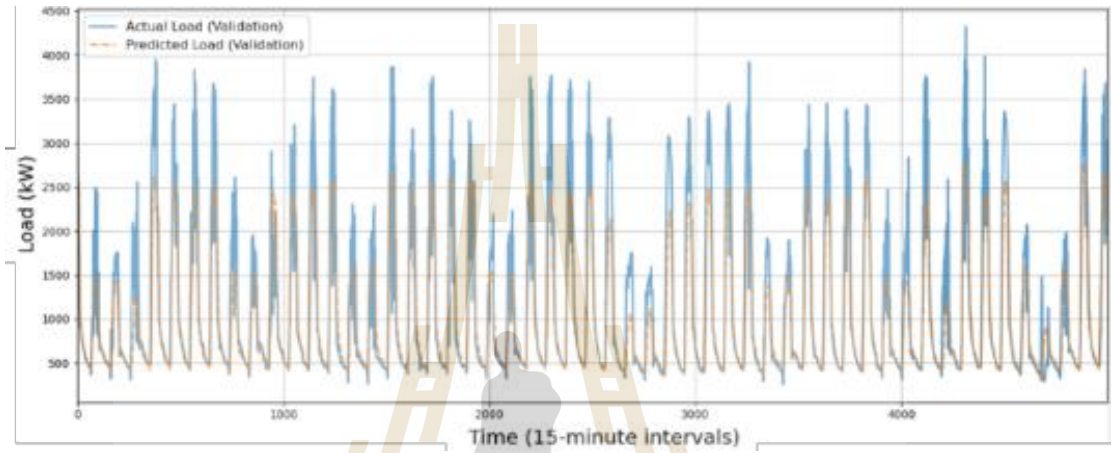
To evaluate the performance of the developed daily electrical load forecasting model in this research, the RNN family models, namely LSTM, GRU, Bi-LSTM, and Bi-GRU, were tested on the real load dataset to compare the model accuracy in the validation phase. In addition, the model training time was recorded to compare the time efficiency between different architectures. The results are shown in Table 3.3, reflecting the forecasting capability of each model under the same context and supporting the selection of the most suitable model in terms of accuracy and performance.

Table 3.3 Evaluation results of model performance in validation metrics and model training time.

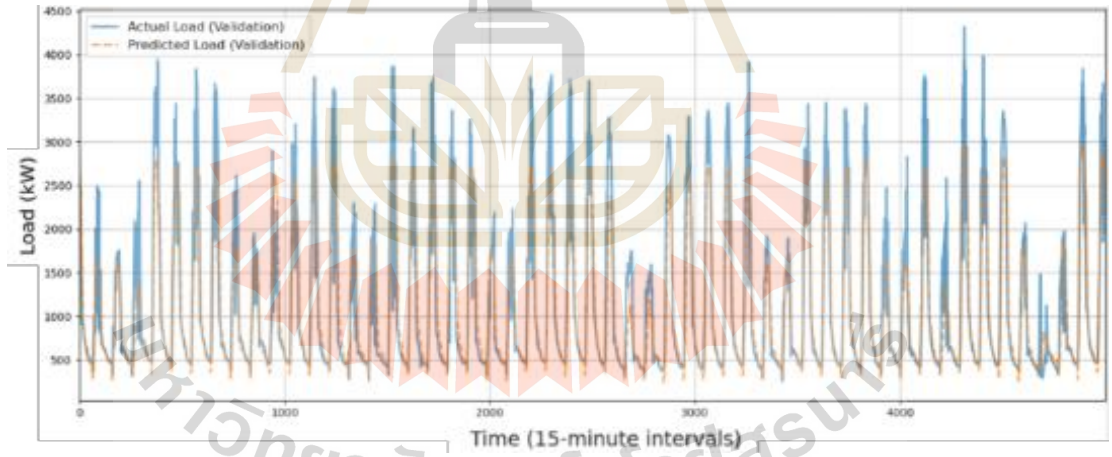
Methods	Validation Metrics			Model Training
	MAPE (%)	MAE (kW)	RMSE (kW)	Time (s)
LSTM	16.36	227.48	428.84	296.34
GRU	20.61	271.96	464.28	176.76
Bi-LSTM	16.86	241.22	439.58	589.14
Bi-GRU	17.71	252.18	453.94	318.95

The GRU model trained faster than the LSTM model, which took 296.34 seconds. Along with the evaluation of the model's performance in the validation set data, the model training time could indicate the device's usage of resources. The findings of the model's electrical load forecasts will be validated using this data. With MAPE = 16.36%, MAE = 227.49 kW, and RMSE = 428.85 kW, the LSTM model performs best after the performance evaluation. The Bi-LSTM model, with MAPE = 16.86%, MAE = 241.22 kW, and RMSE = 439.5889 kW, is the experimentally efficient model. The two models' training errors are comparable to those of the GRU model (MAPE = 20.61%, MAE = 271.96 kW, and RMSE = 464.29 kW) and Bi-GRU (MAPE = 17.71%, MAE = 252.19 kW, and RMSE = 453.94 kW), according to the three metrics. According to the validation dataset, results demonstrate the extent to which the LSTM and Bi-LSTM models learn

and capture relationships. With its speed and predicting accuracy, LSTM offers potential for real-world use. Table 3.3 displays the values in the validation set, which includes the model training settings or the simulation results of the validation metrics' performance.



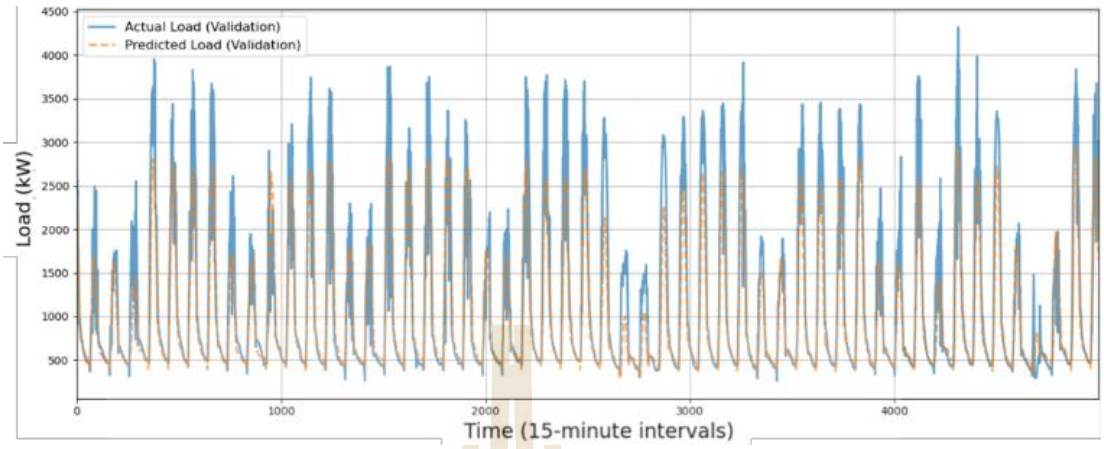
(a)



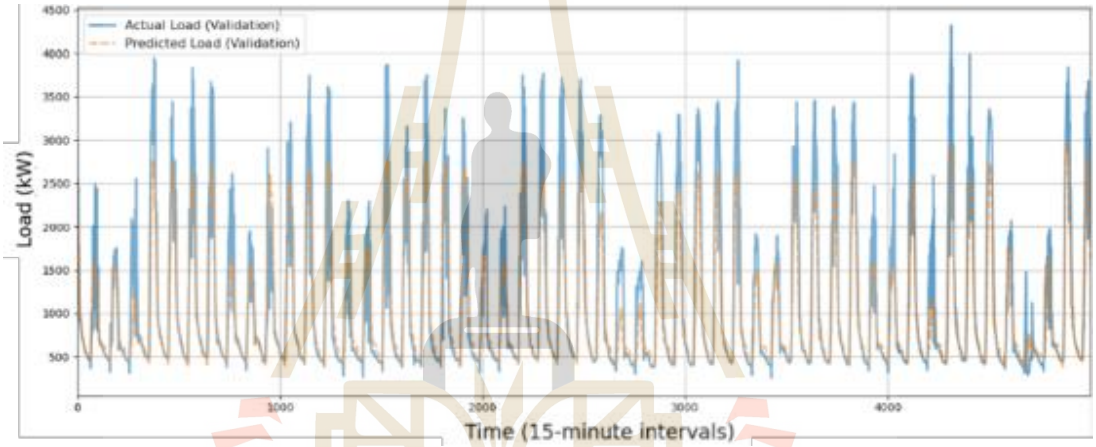
(b)

Figure 3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models

มหาวิทยาลัยเทคโนโลยีสุรนารี



(c)



(d)

Figure 3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models

Fig. 3.6 presents the prediction results between the actual load and the model's predicted load on the validation data. It includes the predictions of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. The results of the prediction align with the mentioned earlier metrics.

3.8.3 Performance of each model on weekday and weekend loads

After model training and validation, the model is tested in real use. The test set data is divided into two types: weekday load is the load from Monday to Friday, which has a higher total electricity demand than the holiday load. Additionally,

weekend load is the load on Saturday or Sunday with the largest variability in electricity demand. The following test days have been selected to evaluate the precision and capacity for forecasting each load model when the electrical load exhibits distinct characteristics: The weekday load is Wednesday, March 1, 2023. And the weekend load is Sunday, October 1, 2023. Forecasting the electrical load output for a day or 24 hours in the future is required, with data being displayed every 15 minutes. There will be a total of 96 data sets. The actual test date is simulated through each model. The results obtained from simulating the weekday load are shown on Fig. 3.7 and the results from simulating the holiday load are shown on Fig. 3.8. The forecast graph will appear.

The blue line represents the actual electrical load, or pure load, the yellow line, the green line, the red line, and the purple line. Shows the forecast of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. In the weekdays load forecast, the LSTM model graph gives the forecast that is closest to the actual load graph. The graph is closely related to the actual load graph. The next order has the forecast that deviates from the actual load, which are the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively. However, other models can provide the forecast of the actual load trend. In the part of the weekend's load forecast, the LSTM model graph still gives the forecast that is closest to the actual load graph. However, there are still some load periods that do not give the forecast that is as close to the actual value as it should be because it is a day with a significant load fluctuation. Overall, the LSTM model still performs well. The models whose forecast graphs are closest to the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively.

มหาวิทยาลัยเทคโนโลยีสุรนารี

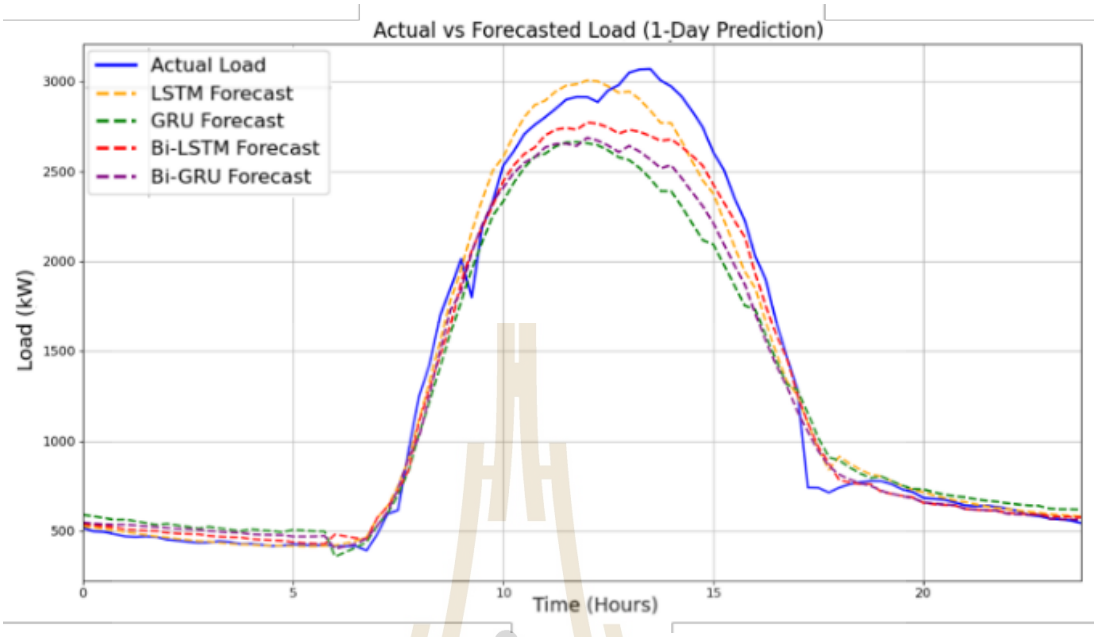


Figure 3.7 One-day-ahead electricity load forecast for each model on weekdays

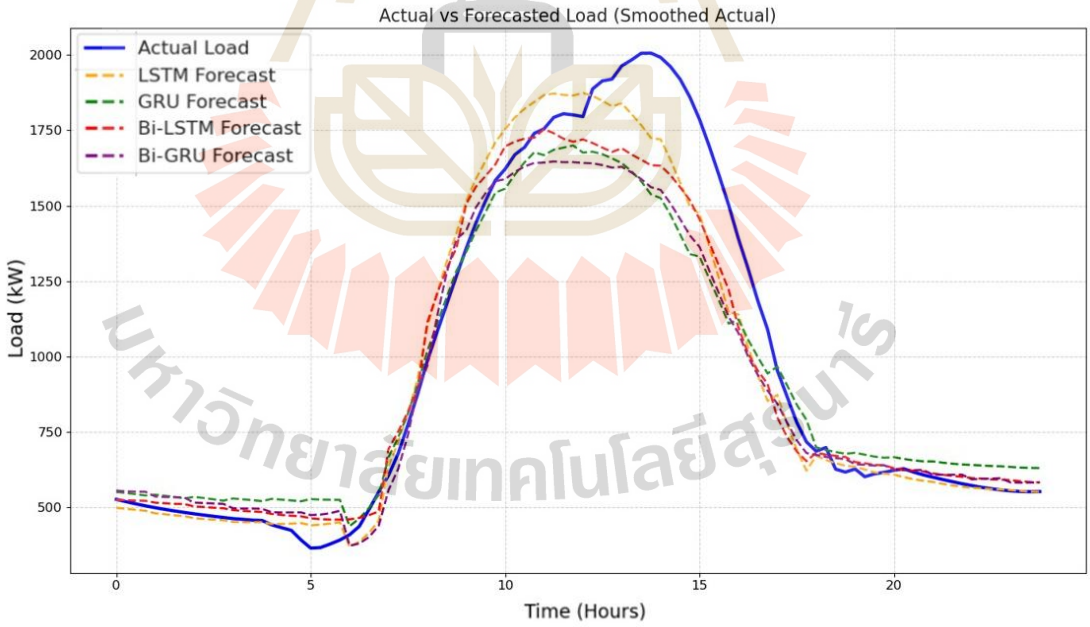


Figure 3.8 One-day-ahead electricity load forecast for each model on weekends

Table 3.4 Performance of each model on weekday load

Methods	Model Performance of Weekday Load		
	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	**6.09**	**81.51**	**122.84**
GRU	13.29	170.33	242.46
Bi-LSTM	7.38	91.38	131.30
Bi-GRU	9.88	135.16	196.28

Table 3.5 Performance of each model on weekend load

Methods	Model Performance of Weekend Load		
	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	9.62	115.12	214.53
GRU	15.51	157.18	254.17
Bi-LSTM	11.21	127.00	219.68
Bi-GRU	11.42	138.39	243.42

In addition, the performance of each model was evaluated through all three indicators or through evaluation metrics. The results of the weekday load evaluation are as follows: The LSTM model has a MAPE = 6.09%, MAE = 81.51 kW, and RMSE = 122.84 kW. The GRU model has MAPE = 13.29%, MAE = 170.33 kW, and RMSE = 242.46 kW. The Bi-LSTM model has a MAPE = 7.38%, MAE = 91.38 kW, and RMSE = 131.30 kW. And the Bi-GRU model has a MAPE = 9.88%, MAE = 135.16 kW, and RMSE = 196.28 kW. Shown in Table 3.4. The results of the holiday load evaluation are as follows: The LSTM model has a MAPE = 9.62%, MAE = 115.12 kW, and RMSE = 214.53 kW; the GRU model has a MAPE = 15.51%, MAE = 157.18 kW, and RMSE = 254.17 kW; the Bi-LSTM model has a MAPE = 11.21%, MAE = 127.00 kW, and RMSE = 219.68 kW; and the Bi-GRU model has a MAPE = 11.42%, MAE = 138.39 kW, and RMSE = 243.42 kW. Shown in Table 3.5. From the evaluation of the model performance, the results of the weekday load, the LSTM model shows the lowest forecast error when compared to the Bi-LSTM model, Bi-GRU model, and GRU model. For the results of holiday load,

the LSTM model still has the lowest error of all metrics when compared to other models.

However, it still has a higher error than the weekday load because it is a day with high volatility, especially in hours 13 to 18, which has the highest volatility, resulting in the highest forecast error during that period. Overall, the LSTM model still has the best power demand prediction and gives the lowest error for that day. The results show that the most efficient and accurate model for university load data is the LSTM model because the model can learn the patterns and relationships of long-term periods well, including managing the time series data in the data set most efficiently. The weekday load pattern has slightly higher forecast accuracy than the holiday load because the holiday load is more volatile. The next most efficient models are the Bi-LSTM and Bi-GRU models because the type of data that needs to be forecasted may not give enough weight to consider the data from the past and the future, causing the model to learn the relationship and not be able to use its full potential. The last model is the GRU model because it is a simpler model than the LSTM model, which makes it not as good at learning the relationship of the data set, and this type of data may not be suitable for the model. While Suranaree University's power load can be reliably predicted by LSTM and Bi-LSTM models, there are still some restrictions that should be considered when using these models in other situations.

To properly understand long-term correlation patterns, the models need a lot of high-quality data. In the case of data-constrained environments, such as small-scale power systems in rural areas or organizations with data shortages, the model performance may be reduced. In addition, Bi-LSTM and Bi-GRU models are complex and require higher computing resources than GRUs, which may not be suitable in systems with limited computing resources, such as IoT devices or edge computing systems. However, the models can be applied in different environments, both in the large-scale energy industry and in distributed energy systems, by adjusting the model structure and parameter settings to suit the data complexity in each context or adding variables such as weather data or sunrise-sunset times to increase the accuracy of predicting fluctuating energy usage. The use of GRU models with higher model training speed in real-time systems may be a suitable option. The results of this

study have important implications for the energy sector in many dimensions. The developed electricity load forecasting model can help enhance the efficiency of energy management strategies in various contexts. By precisely predicting energy demand ahead of time, energy suppliers may effectively plan power generation and distribution, minimize overloading, and minimize transmission line losses through integration with smart grid energy management technologies. These models can be used in industrial energy management to predict the energy demand for pieces of equipment or production processes. This helps energy managers plan when to use energy (demand-side management), lower electricity costs during periods of high energy prices, and make the production system more sustainable. In addition, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources such as solar and wind power, which can help reduce energy imbalances and improve the efficiency of renewable energy use, as well as support the reduction of unnecessary energy use as part of the strategy to reduce greenhouse gas emissions in the energy sector.

3.8.4 Performance of LSTM model on 12 test sets

In power forecasting for systems with continuous load variability, the use of machine learning models, especially Long Short-Term Memory (LSTM) models, is a popular approach that shows good performance in many contexts. This is because LSTMs have the ability to remember data sequences and detect long-term patterns in data. Testing the LSTM model on datasets with different characteristics in terms of energy consumption behavior, such as holidays, exam days, or different seasons, can help to clearly assess the model's flexibility and accuracy. The datasets were managed by performing feature engineering before importing the model. Therefore, the analysis in this section focuses on comparing the performance of the LSTM model on 12 test datasets, which reflect different temporal characteristics, along with the use of performance indicators such as MAPE, MAE, and RMSE. The performance evaluation results of the LSTM model on 12 test datasets, each of which represents a variety of time periods or situations, such as long holidays, school openings, exam days, or

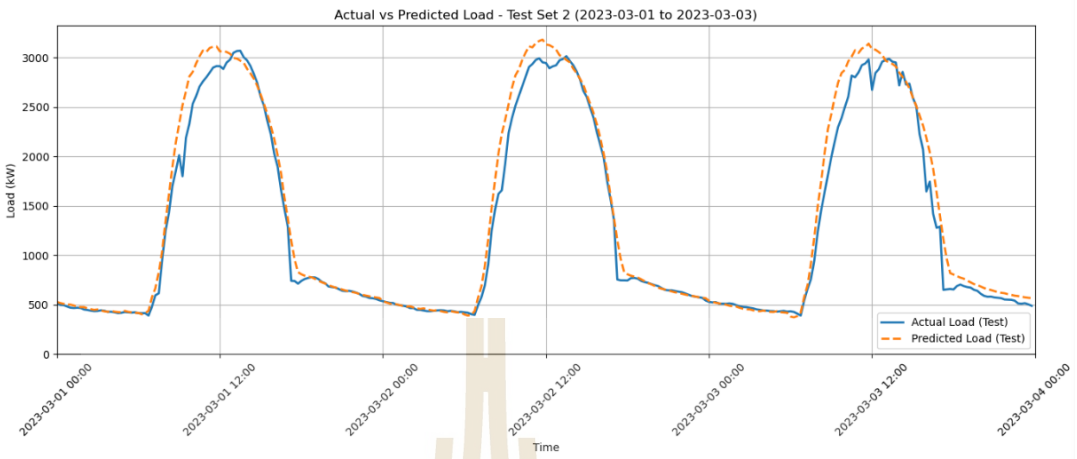


Figure 3.9 Forecasting Result – Test Set 3: Exam Period

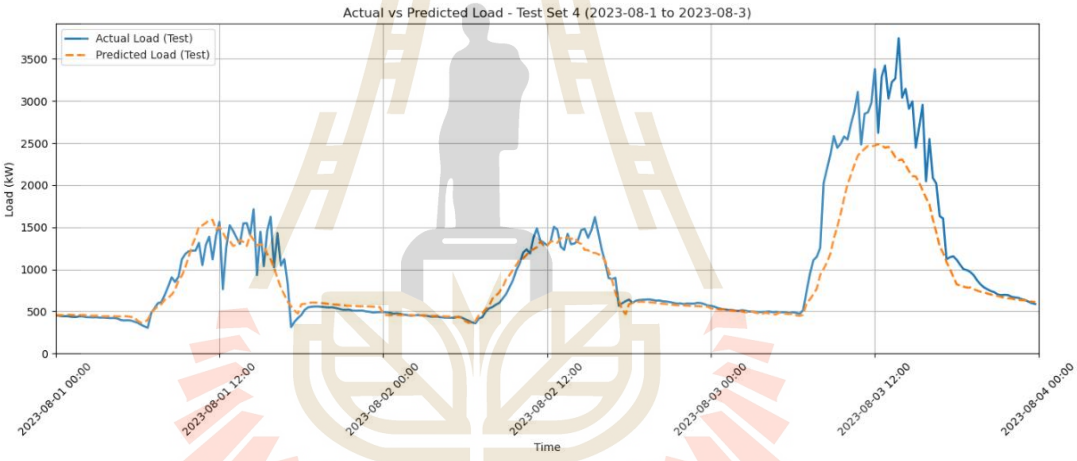


Figure 3.10 Forecasting Result – Test Set 6: Start of the Semester

different seasons, are presented. Each data set consists of 3 days of data to clearly show the trends in energy consumption behavior in each period, The results are shown in Appendix A.

From the evaluation results, it was found that Set 3 (exam period 2023-03-02) gave the lowest MAPE value at 5.93%, indicating the highest forecast accuracy with MAE equal to 83.05 kW and RMSE equal to 145.60 kW, which showed good results in terms of percentage of error. Meanwhile, set 6 (the start of the semester, 2023-08-02) gave the lowest MAE and RMSE values, namely MAE = 59.18 kW and RMSE = 93.66 kW, indicating that in terms of absolute error and root mean square error, the model

can predict the closest to the actual value. For analysis, the forecast results of Set 3 and Set 6 are presented in graph form. The graph shows the difference between the actual value and the value predicted by the model, helping to show the level of model consistency in each period shown in Fig. 3.9 and Fig. 3.10.

Table 3.6 Model Performance on Validation and Average Performance on 12 Test Sets.

Methods	Model performance of the validation set			Average performance of the model on 12 test sets		
	MAPE (%)	MAE (kW)	RMSE (kW)	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	13.71	177.58	315.67	14.07	188.77	302.04

In addition to the evaluation of each dataset, Table 3.6 shows the overall evaluation results in two aspects: the prediction results on the validation set and the average model results from testing on the 12 datasets. On the validation set, the LSTM model yielded MAPE = 13.71%, MAE = 177.58 kW, and RMSE = 315.67 kW. The average of the model on the 12 test data sets yielded MAPE = 14.07%, MAE = 188.77 kW, and RMSE = 302.04 kW. These results reflect that the model can maintain a stable level of accuracy when comparing the validation set and the test set, with an acceptable level of error for use in short-term forecasting systems. From the evaluation of the LSTM model's performance on various data sets in terms of time and context, it was found that the model can forecast accurately in some time periods, especially during exams and school openings, where the forecasting results are more accurate than other periods. The use of lag features and clustering data by time characteristics enhances the model's potential to capture the patterns of load data. In addition, the overall evaluation of the validation set and the average values from the test sets confirm that the model is stable and can be effectively applied to load forecasting in various contexts.

3.9 Summary of the daily forecast

This article presents an alternative approach to developing short-term electric load forecasting models and improving their accuracy through deep learning techniques, including the Day-Aware Data Arrangement method. The most widely used forecasting methods in this field were simulated and compared to evaluate their performance and accuracy. The daily electric load forecasting for Suranaree University of Technology often correlates with past time intervals, allowing LSTM to effectively capture these patterns better than other models. As a result, its load predictions are the closest to the actual values. Adding bidirectional layers to the models did not significantly enhance the results for electric load data. The GRU model, while easier to implement and faster to process, demonstrated lower performance and accuracy compared to the LSTM model for sequential data of this nature. Notwithstanding the encouraging outcomes, this study also discusses significant shortcomings and challenges for further applicability. For maximum performance, high-quality and adequate data are necessary, which might be difficult in contexts with limited resources. Nonetheless, similar models can be used in various energy sectors by reorganizing them, adding new variables, and adjusting the parameter values. The study's findings have important ramifications for the energy industry overall. Planning energy production and distribution effectively, reducing energy losses, and maximizing the integration of renewable energy sources are all made possible by the inclusion of accurate forecasting models, particularly LSTMs, into energy management techniques. Faster GRU models may be suitable for real-time applications where computational resources are limited. And adding Feature Engineering to evaluate the stability of the LSTM model on 12 test sets to study the variability of the results that may be caused by different data characteristics at different times.

CHAPTER IV

Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus

4.1 Introduction

This section proposes an approach to developing a one-day-ahead electrical load forecasting model using deep learning techniques. It compares deep learning models with basic models, namely Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM, to improve accuracy and reliability through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis. This approach utilizes data collection and feature engineering to optimize the context of an electrical load with integrated solar power generation, Using Suranaree University of Technology (SUT) campus electrical system as a case study. Three accuracy indices are examined: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). Among the NN-based models, Bi-LSTM and LSTM tend to provide the best overall forecasting performance from the input data, and the Bi-LSTM model is the most accurate model with the lowest metric value among all models. The lag load feature, or the value of the electrical load in the past one day, is the feature that is most related to the forecasting target and has the most impact on the model's decision. Parts of this thesis have been published in ECTI Transactions on Electrical Engineering, Electronics, and Communications (2025), is shown in the Appendix D, and another set of simulated data is presented to further enhance the reliability of the framework, also shown in this section.

4.2 Data Collection and Preparation

Before entering the model for load forecasting, Suranaree University of Technology's electricity consumption data was collected for 1 year, collected every

15 minutes in 2023, totaling 35,040 data sets, which is considered high-resolution data for load forecasting. The data was collected from Automatic Metering (AMR) maintained by the Provincial Electricity Authority (PEA). In addition, solar energy generation data was collected from an Automatic Metering (AMR) maintained by an external company, ENOS, with the same resolution as the electricity load data. Weather data and academic calendar data for that year were collected and prepared to input all this data into the future feature engineering method. The required data is the actual load data. Therefore, the load obtained from the meter was combined with the solar energy generation data to obtain data showing the actual load of the university.

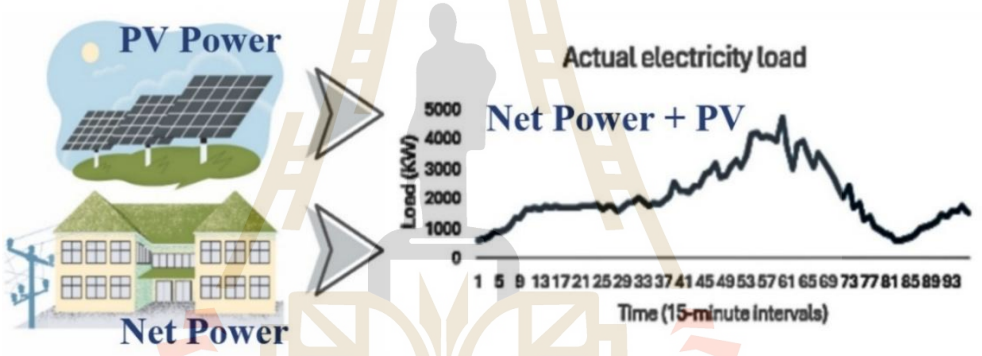


Figure 4.1 Preparation of Actual Electricity Load Data

The preparation of the real electricity load data can be shown on Fig. 4.1. All data used as features and output for the model to learn were divided into two parts: the first part was the data set to be trained, and the other part was partitioned for model validation. Time Series Cross-Validation (TSCV) was performed on this data set, which is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. The training and validation sets were generated using a Time Series Split with 5 folds. The second part was the data set tested on the model after training and validation were completed. It was a data set that the model had never seen before, so the results were unbiased and divided into 4608 data points. For testing, this group of data will be divided into 10 sets; each set

will have 3 days. Each set is divided according to the types of data groups, namely long holiday, weekdays during the school term, exam period, summer season, the start of the semester, compensation holiday period, rainy season, weekdays and weekends, term break, and winter season.

4.3 Feature Modeling and Representation of Electrical Load Patterns

Feature engineering has a group of data that has the characteristics of feature creation, which is creating new features from existing time data and converting time data that may not be easy for the model to logical data with contextual meaning.

4.3.1 SUT's Campus Data

It consists of the following techniques: In the feature transformation method, features that use this method include Weekdays and Weekend, Daytime and Nighttime, Compensatory Holiday, and Long Weekend, which are converting to variable values and binary representation with numbers 0 and 1. In the time-based feature engineering method, features that use this method include yearly hours, which divides the data into meaningful time sequences with 1 year's data every 15 minutes, creating features that allow the model to learn the day and time. The Time Component Extraction method features that use this method include Quarter-Hourly Interval, which extracts time components into separate features, creating new features that describe the sub-patterns that occur in each hour and the lag features method. Features that use this method include Lag Load, which creates features from previous values by taking historical load values to create features. In addition, there is a group of characteristics: external features. External variables affect the forecast of the target variable, such as ambient temperature, an external factor that directly affects energy use behavior, and dew point, an indicator of air humidity.

4.3.2 New England (ISO-NE) Data

Time-based feature generation plays a crucial role in enabling the model to efficiently learn periodic patterns and long-term trends of electrical load in the second dataset. Additional relevant features can be generated as follows:

The "Day" feature is used to reflect the order of days within a month, allowing the model to recognize changes in electricity usage behavior that may occur at the beginning, middle, or end of the month, such as during payday or before long holidays. The "Month" feature represents the seasonality of electrical load, particularly the impact of different weather conditions throughout the year, such as summer or rainy season, which directly influences electricity demand. The "Year" feature plays a role in reflecting long-term trends in electrical load, such as increases in the number of electrical appliances, infrastructure expansion, or changes in energy policy. This allows the model to appropriately differentiate electrical load behavior over several years. Furthermore, the "Weekday" feature is used to distinguish between weekdays and holidays, a key factor clearly influencing electricity usage patterns. Generally, electrical load patterns on weekdays differ from holidays, both in terms of load levels and peak times. Incorporating this feature allows the model to better learn about the differences in electricity usage behavior across different types of days.

In summary, adding time-related features—Day, Month, Year, and Weekday—along with Lag Load, Ambient Temperature, and Yearly Hours features in the ISO-NE dataset for further simulation enhances the model's ability to capture temporal relationships, long-term trends, and periodic patterns of electricity load—critical components for accurate and stable electricity load forecasting in the context of time series data.

4.4 Feature Selection and Mutual Information

Mutual Information (MI) captures the relationship of input features to a non-linear data model, shows their importance, and indicates how the feature reduces uncertainty with respect to the forecasting target. It is used as a tool for selecting appropriate forecasting features, especially in tasks where the number of features varies without necessarily being linear. It is calculated using the following equations (4.1).

$$i(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \cdot log\left(\frac{p(x,y)}{p(x) \cdot p(y)}\right) \quad (4.1)$$

where Y is the target variable, i.e., the future load, X is the input variable, $p(x, y)$ is the joint probability of x and y, and $p(x)$ and $p(y)$ are the marginal probabilities of each variable. The value of MI is expressed as a real number greater than or equal to 0. The higher the value of MI, the more the feature is related to the forecast value. MI can help reduce the size of the data to control both the speed of training and the interpretation of the model. MI is a powerful tool for feature selection for time series models, especially in complex load forecasting tasks that change over time and in different contexts. Selecting appropriate features using MI can make the model more accurate and easier to interpret.

Feature selection in load forecasting each feature plays a physical or behavioral role in load patterns. The Lag Load feature was selected because electrical loads are similar and have daily activities that often repeat within 24 hours, making it a very important representative of daily patterns in forecasting. The Yearly Hours and Quarter-Hourly Interval features help the model understand the time sequence of each day and allow the model to recognize period differences. When used in combination with other features, the model can better learn about daily trends. The Weekdays and Weekend features were selected because electricity usage by electricity consumers differs significantly during holidays and weekdays. The Daytime and Nighttime features were selected because people's behaviour shows higher electricity usage in the morning than at night, so the model should be able to see this difference. The Compensatory Holiday and Long Weekend features have different load patterns during long holidays and special holidays. The Ambient Temperature and Dew Point features are weather-related features that reflect the characteristics of electricity consumers when temperatures and humidity change. They also provide the model with sufficient forecasting capabilities when there is load variation caused by changing weather conditions.

4.5 Modeling Approaches and Baseline Methods

Accurate short-term electricity load forecasting requires models that can effectively capture temporal dependencies, nonlinear relationships, and variations in

consumption patterns under different operating conditions. To comprehensively evaluate forecasting performance and demonstrate the effectiveness of advanced deep learning techniques, this study investigates both baseline forecasting methods and a range of time-sequence deep learning models.

The models used to simulate results in the Recurrent Neural Network (RNN) category were compared and selected. The models with the highest potential for accuracy were LSTM and Bi-LSTM, as discussed in the previous chapter. The scale was then expanded to compare with other categories to cover a wider range of subjects. In the classical statistical time-series method category, the moving average method was selected. This is a traditional statistical forecasting method that calculates the average of the electrical load over a given period and uses that value as a forecast for the next period. It uses the average of the past 3 days. This method does not involve model learning and is used as a basic method. In the Feedforward Neural Network (FNN) category, the DNN model was selected. This is a multi-layered hidden neural network where data flows forward in only one direction from input to output, without memory of the past like RNNs. In the hybrid model category, the models selected were Attention-LSTM and CNN-LSTM.

4.5.1 Attention-LSTM Model

Attention-LSTM consists of an LSTM model that extracts deep features from time-dependent data and an attention component that determines which data in the sequence has the greatest impact on load forecasting. Features fed as input are given into the LSTM model, with the input data at each time point represented by variables X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1}, and X_t. The model then generates a hidden state (h_t) that stores in-depth information about past electrical load behavior, as shown by the equation (4.2).

$$h_t = LSTM(X_t, h_{t-1}, C_{t-1}) \quad (4.2)$$

Each timestep yields the results h_{t-4}, h_{t-3}, h_{t-2}, h_{t-1}, ht, where C_{t-1} is the cell state capable of retaining information for a long period. Since the LSTM model focuses only on the last ht, an attention mechanism is added to emphasize (weight) each hidden

state. It creates weights (α_1, α_2, α_3, α_4, α_5) indicating the importance of the information in that timestep and its impact on the forecast for each hidden state. A softmax function is then used to normalize the result so that the sum equals 1. This is shown in the equation (4.3).

$$\alpha_i = \frac{exp(score(hi))}{\sum_{j=1}^{T} exp(score(hi))} \tag{4.3}$$

Then, the hidden states are multiplied by the weight (α) and combined to create a Context Vector (C), which is the central value summarizing the historical load data that the model prioritizes most. This represents sequence data all the time and can be expressed as the follows equations (4.4).

$$C = \sum_{i=1}^{T} \alpha_i h_i \tag{4.4}$$

Once C is obtained, it is combined with the LSTM Output to create the Feature Final (Fea_{final}), which represents the features processed by both LSTM and Attention. It is then passed through the Dense Layer to predict the electrical load for the next day. The attention mechanism was implemented using Luong dot-product attention, where the attention weights (α) are normalized using the softmax function. Feafinal represents the combined feature vector generated from the LSTM hidden states and attention mechanism. The structural diagram of the model is shown in Fig. 4.2.

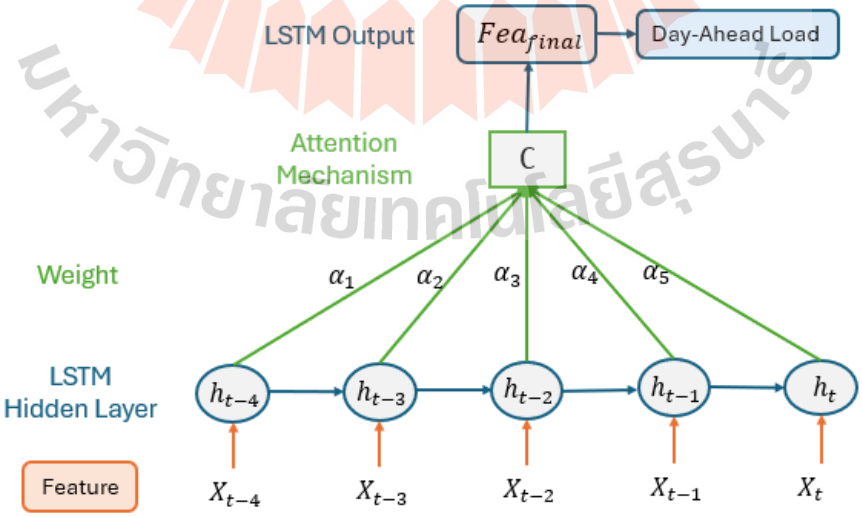


Figure 4.2 Attention-LSTM architecture

4.5.2 CNN-LSTM Model

The CNN-LSTM model is a combination of Convolutional Neural Network (CNN) and LSTM models for improved feature extraction and learning of temporal dependencies. CNN extracts feature from sliding window load data using 1D convolution to compute the feature map at position t. A kernel of size k is used, employing multiple filters to detect different patterns, as shown in the equation (4.5).

$$z_t = \sum_{i=0}^{k-1} w_i \cdot x_{t+i} + b \tag{4.5}$$

Where w_i is the filter weight (kernel weight), b is the bias, i is the index of the position within the kernel, x_{t+i} is the model input at time $t+i$, and z_t is the feature value obtained after convolution. A pooling layer is used to reduce the dimensionality of the data and extract only the important features, where p is the size of the pooling window, as shown in the equation (4.6).

$$p_t = max(z_{t:t+p}) \tag{4.6}$$

Where p_t is the pooling result at time position t, $z_{t:t+p}$ is the sequence of feature map values obtained from the convolution step of the CNN from time t to $t+p$, and $max(\cdot)$ is the max pooling function. After the CNN model finishes extracting features, it sends the obtained features to the LSTM model to learn the time sequence and finally sends them to the Dense Layer to forecast the electricity load for the next day. The predicted load for the next day is given by the equation (4.7).

$$\widehat{Y_{t+1}} = W_d \cdot H_t + b_d \tag{4.7}$$

This is the step for creating the output layer of the model, where $\widehat{Y_{t+1}}$ is the value, the model predicts at the next time point $t+1$, and H_t is the hidden state of the LSTM at time t, which is the result of the time sequence learning process. This data has first featured using a CNN for short-term spatial or pattern extraction before being fed into LSTM to learn temporal relationships. It represents temporal memory that includes past load patterns and the influence of other features. W_d is the weight matrix of the Dense layer (fully connected layer), which transforms the hidden state of the LSTM into a dimension suitable for the desired output prediction, and b_d is the bias of the output layer, which adjusts the prediction value independently of the input,

allowing the model to be more flexible in estimating the true value. The CNN-LSTM model uses CNN to extract key patterns from historical data, then uses an LSTM to learn temporal relationships, and finally uses the Dense layer to transform the hidden state into the predicted electrical load value.

4.5.3 DNN Model

As part of the Artificial Neural Network (ANN) method, it learns deeper input-output relationships, especially in tasks with nonlinear data. The model consists of three parts: the Input Layer, which receives feature data and prepares it for encoding before sending it to the next layer; next are the Hidden Layers, which learn increasingly complex data and can be divided into multiple layers to accommodate more complex incoming data; and finally, the Output Layer, which predicts the continuous load (h^l). The calculation is shown in the equation (4.8).

$$h^l = f(w^l \cdot h^{l-1} + b^l) \tag{4.8}$$

Where w^l is the weight of the l layer, b^l is the bias constant, and $f(\cdot)$ is the activation function.

4.5.4 Moving average

This is a traditional, basic method used to forecast electrical loads with relatively stable patterns and minimal data fluctuations. The type used is the Simple Moving Average (SMA). Future forecasts depend on the average load usage over similar past periods, focusing on patterns and trends. The forecast value for the current day at time t will be equal to the average of the past three days at time t, as shown by the equation (4.9).

$$\widehat{Y_{d,t}} = \frac{Y_{d-1,t} + Y_{d-2,t} + Y_{d-3,t}}{3} \tag{4.9}$$

Where $\widehat{Y_{d,t}}$ is the forecast value for day d at time t and $Y_{d-1,t}$ is the actual load value for the previous days at time t.

4.6 Time Series Cross-Validation (TSCV)

Model performance evaluation is an important part of the data analysis process, especially in tasks involving time series data such as load forecasting, which are characterized by temporal order and temporal dependency, making it impossible to use traditional cross-validation techniques that are applied to independent and identically distributed data. To solve this problem, Time Series Cross-Validation (TSCV) is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. In conventional cross-validation, such as K-Fold Cross-Validation, data is randomly partitioned regardless of the order of the data. This will result in future data being present in the training set and historical data being present in the test set, resulting in data leakage that makes the model evaluation results do not reflect the true performance. TSCV solves this problem by preserving the order of the data and using a "walk forward" or "rolling window" approach instead. Here we will discuss the expanding window (walk-forward validation) approach, which starts with a small training data set and gradually adds new data to the training set, while the test set always moves in time. This is suitable for models that need to "learn" from the past and predict the future, especially in contexts where the data tends to change cumulatively, as shown in Fig. 4.3.

Fold	Train Index	Validation Index
1	[0, 1]	[2, 3]
2	[0, 1, 2, 3]	[4, 5]
3	[0, 1, 2, 3, 4, 5]	[6, 7]
4	[0, 1, 2, 3, 4, 5, 6, 7]	[8, 9]

Figure 4.3 Expanding Window

มหาวิทยาลัยเทคโนโลยีสุรนารี

Time Series Cross-Validation is an important process that enables accurate model evaluation in the context of temporal data, especially in load forecasting applications where data are time-varying and have a seasonal structure. Proper selection of TSCV prevents data leakage, increases the reliability of evaluation results, and supports the creation of models that can be confidently put to practical use.

4.7 Shapley Additive Explanation (SHAP)

In a load forecasting system that uses complex models, users often need not only accurate predictions but also to understand why the model produces those predictions in order to build confidence, support business decisions, and verify the stability of the model when the environment changes. SHAP is one of the tools that can provide a rational and easily interpretable explanation of these models.

SHAP is a method based on the Shapley Value from Cooperative Game Theory, which measures how each feature contributes to the model's decision. SHAP transforms the model description problem into a game in which each feature is a player, and the model's outcome is a reward that must be shared (prediction). The Shapley Value calculation considers all subsets of features (feature coalitions) to assess the impact of each feature. The calculation principle (conceptually) can be described as the Shapley value of feature i is defined by the following equation (4.10).

$$\emptyset i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)] \qquad (4.10)$$

where N is the set of all features, S is the subset of features excluding i, $f(S)$ is the predicted value of the model when using only the feature in S, and $\emptyset i$ is the average of the effects that feature i has on the predicted value when combined with all other features in various orders.

In load forecasting, the input features can be complex. SHAP can analyze how much historical load (lag features) or temperature influence each time period, interpret which factors cause the model to overproduce or underproduce the forecast, and

identify globally and locally important features, or analyze both overall and case-specific important features. SHAP is the current model explainer, especially for load forecasting tasks that require both accuracy and transparent understanding of the model. It allows us to look into the black box and distinguish how each feature affects the outcome, which is in line with the goal of explainable Artificial Intelligence.

4.8 Forecasting Process Description

The development of an accurate and interpretable short-term load forecasting model involves a series of systematic processes, each contributing to the robustness and practicality of the final solution. This research was carried out in multiple stages, starting from the preliminary design and comparison of deep learning models and progressing through advanced techniques for feature selection, model validation, and explainability. The overall workflow can be divided into three main phases, each emphasizing different aspects of model enhancement: initial development, performance refinement, and interpretability.

Phase 1: Initial Development and Model Comparison The research began with data collection and preprocessing, focusing on historical load data from Suranaree University of Technology, complemented by contextual variables such as temperature, humidity, calendar effects, and time indicators. The raw data were cleaned, normalized, and structured into a time series format suitable for model training. Subsequently, feature engineering was employed to capture the temporal characteristics of the dataset. Manually crafted features, such as day of the week, hour of the day, holiday indicators, and seasonality factors, were added to enrich the dataset. These features are known to have a significant impact on electricity consumption in institutional settings. The core of this phase was the design and training of four deep learning models. These models were selected for their proven effectiveness in modeling time-dependent patterns. After training the models on the same dataset, their performance was evaluated using standard metrics, including Root

Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The comparison results consistently show that this model outperforms other models, especially during periods of high workload volatility. Therefore, it is selected as the most suitable model for further development.

Phase 2: Feature Selection and Time-Aware Validation To improve accuracy further and address potential overfitting, the next phase introduced automated feature selection using the mutual information (MI) method. MI assesses the non-linear dependency between each input variable and the target load, helping to retain only those features that contribute meaningfully to prediction performance. This step allowed for dimensionality reduction while preserving the complexity needed to reflect real-world electricity usage patterns. After refining the feature set, the model was retrained using the selected features. In this phase, model evaluation was conducted using Time Series Cross-Validation (TSCV) with an Expanding Window strategy. Unlike traditional k-fold cross-validation, which can lead to data leakage in time series data, TSCV respects the temporal order of the data, simulating real-world forecasting conditions where only past information is available to predict the future. This enhanced validation approach provided a more realistic and robust assessment of the model's generalization ability.

Phase 3: Explainability and Model Transparency Although deep learning models can offer high accuracy, their black box nature often limits their practical utility in energy decision-making environments. To address this, the final phase introduced an explainability mechanism using SHAP. SHAP quantifies the contribution of each feature to a specific prediction (local explanation) and identifies the most influential features across the dataset (global importance). Insights from SHAP analysis were used to refine the feature set further. Features with low or inconsistent importance were reviewed, and the model was retrained with a final set of contextual and statistically relevant variables. This interpretability step not only improved the transparency of the forecasting system but also provided actionable insights for energy managers and

stakeholders, who could now better understand the reason behind the model's outputs.

The complete workflow—beginning with basic data preparation, progressing through model comparison, and culminating in interpretability—forms a robust framework for load forecasting that balances accuracy, transparency, and practical applicability. Each phase is built upon the previous one: initial model comparisons identify a strong baseline, advanced feature selection and time-aware validation enhance accuracy, and SHAP-based interpretability ensures model decisions can be trusted and understood. The overall research process is presented in Fig. 4.4, which illustrates the step-by-step process from data collection to descriptive load forecasting.

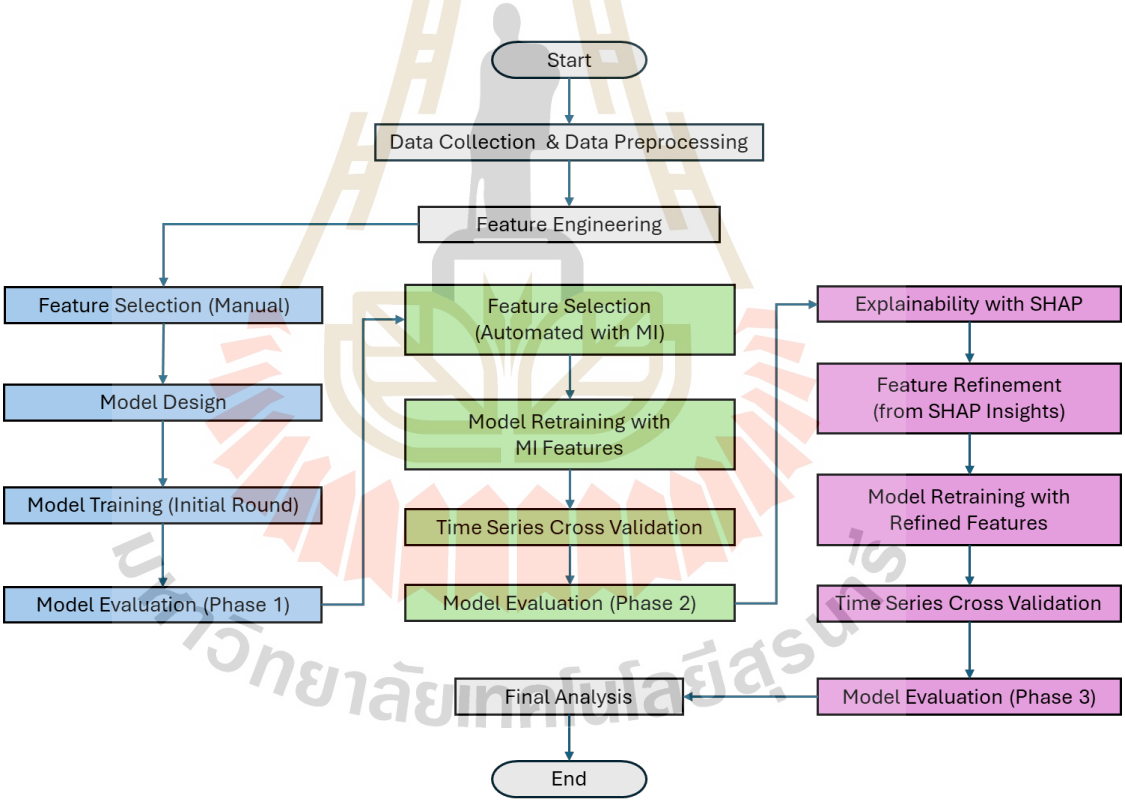


Figure 4.4 Process Description

4.9 Load Forecasting Framework

Electrical load forecasting requires the preparation of essential data, including aggregation and collection from reliable sources in the forecasting area, focusing on actual electricity load data from meters. This data is then fed into feature engineering. Two types of data are used: feature generation, which converts raw data into numbers that are easier for the model to understand and allows the model to better learn about data differences, and external feature data, which learns about data trends. This process takes place during the data collection and preparation process.

The next step in the simulation process is automatic feature selection using MI to assess the nonlinear relationship between each input variable and the target load. This helps preserve features that are important for forecasting. This reduces data complexity while retaining the most important features. The data is then grouped into three groups for training, validation, and testing, based on proportions. The resulting data is then fed into the model by designing the model, setting various parameters, and training deep learning models in various categories: Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. After training the models on the same dataset for all models, then, in the results and analysis process, the performance is evaluated using standard metrics: root means square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

To enhance the transparency and clarity of the black-box model, the SHAP tool is used to guide decision-making on feature prioritization. This tool identifies the lowest and highest priority features for the final pre-load forecast. Figure 4.5 shows the entire forecasting process. The first step is the data collection and preparation process. The next step is the simulation process. The last step is the results and analysis process.



Figure 4.5 The Complete Forecasting Framework

4.10 Simulation Results and Discussion with SUT's Campus Data

The electrical load forecast from all the data obtained and the tools mentioned above will be used in simulation using the Python 3 program to show the results comprehensively and explain the accuracy and reliability of the model in the area and context of Suranaree University of Technology. It is a 1-day-ahead load forecast using historical data for 1 year of forecasting in 2023. The details are presented in this section. Using Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models.

4.10.1 Implementation and Experimental Setup

The settings for each model are shown in Table 4.1. All models use similar parameters, namely the Dropout Rate parameter with a value of 0.2 to set the percentage of that unit. To prevent overfitting in the model, the learning rate is set to 0.001 to determine the speed of parameter tuning during learning. The optimizer uses Adam to automatically adjust the learning rate. The batch size, which is the number of data used to calculate the gradient per iteration of parameter tuning, is set to 32. The Epochs, which is the number of iterations to learn the whole data set, is set to 50. The loss function is used to evaluate the model error with MSE. In the different parts, each model is a layer part, which is the basic structure of the model that defines data processing. The DNN model has a Flatten Layer to transform the data and feed it into a Dense Layer, with the number of units in each Dense Layer having the units or the complexity of learning in each layer as 128, 64, and 1. The CNN-LSTM model has Conv1D filters, or the number of filters, as 64. The kernel size is the length of the

repeated pattern detection filter with the value of 3. The pool size for reducing the dimensionality is set to 2. The LSTM Units are set to 100 for learning the temporal relationship after Conv1D has extracted features and has a Dense value of 1 in the model part. The Attention-LSTM has an Attention Layer that prioritizes the timesteps that affect the prediction and returns 100 LSTM Units with a Dense of 1. In the LSTM model, the LSTM Unit Capacity is 100, and the Bi-LSTM model has a Bidirectional Wrapper to process two directions and uses 100 LSTM Units in each direction with a Dense of 1.

Table 4.1 Parameter Setting

Model	Parameters Layer
DNN	Flatten, Dense (128), Dense (64), Dense (1), and Units: 128, 64, 1
CNN-LSTM	Conv1D (64), kernel size (3), MaxPooling1D (pool size=2), Dense (1) and LSTM (100)
Attention-LSTM	LSTM (100), return sequences (True), Dense (1) and Attention
LSTM	LSTM (100) and Dense (1)
Bi-LSTM	Bidirectional (LSTM(100)) and Dense(1)

The selection of the appropriate hyperparameter values for each model was performed using a trial-and-error method. The parameters were adjusted within a specified range and then evaluated by evaluating the minimum MAPE value displayed when each parameter was adjusted within each range. Hyperparameter units were set in a range of 64 to 128, with a best value of 100. The dropout rate was set in a range of 0.1 to 0.3, with the best value of 0.2. The learning rate was set in a range of 0.005 to 0.0005, with a best value of 0.001. Timesteps were set in a range of 4 to 12, with a best value of 8. Batch size was set in a range of 16 to 64, with a best value of 32. Early stopping was set in a range of 5 to 20, with a best value of 10. As shown in Table 4.2, these hyperparameter sets were then used to simulate the results of each model. The details of the parameter adjustment in the Bi-LSTM model, evaluated using the MAPE

indicator, show a minimum value of 17.53% resulting in the optimal parameter range shown in Table 4.3. Example code for simulating the Bi-LSTM model is shown in Appendix B.

Table 4.2 Selecting the appropriate hyperparameter

Hyperparameter	Adjustment range	Best parameter values
Units	[64 128]	100
Dropout Rate	[0.1 0.3]	0.2
Learning Rate	[0.0005 0.005]	0.001
Timesteps	[4 12]	8
Batch Size	[16 64]	32
Early Stopping	[5 20]	10

Table 4.3 Details of parameter adjustment via indicators

Case	Units	Dropout Rate	Learning Rate	Timesteps	Batch Size	Early Stopping	MAPE (%)
1	64	0.2	0.001	8	32	10	21.28
2	128	0.2	0.001	8	32	10	19.69
3	100	0.1	0.001	8	32	10	18.64
4	100	0.3	0.001	8	32	10	21.56
5	100	0.2	0.005	8	32	10	19.95
6	100	0.2	0.0005	8	32	10	19.98
7	100	0.2	0.001	4	32	10	21.01
8	100	0.2	0.001	12	32	10	18.83
9	100	0.2	0.001	8	16	10	19.47
10	100	0.2	0.001	8	64	10	17.72
11	100	0.2	0.001	8	32	5	21.04
12	100	0.2	0.001	8	32	20	19.52
13	**100**	**0.2**	**0.001**	**8**	**32**	**10**	**17.53**

4.10.2 Mutual Information Scores of SUT Data

To prioritize the feature data before importing the model with the MI tool, the relationship of the features and the target electrical load dip data groups is measured as both linear and non-linear data. It shows the score ranking of each feature. The Lag Load feature has an MI score of 1.198, the Yearly Hours feature has an MI score of 0.976, the Daytime and Nighttime feature has an MI score of 0.405, the Ambient Temperature feature has an MI score of 0.378, the Dew Point feature has an MI score of 0.241, the Weekdays and Weekend feature has an MI score of 0.055, the Compensatory Holiday feature has an MI score of 0.018, the Long Weekend feature has an MI score of 0.014, and the Quarter-Hourly Interval feature has an MI score of 0.002, as shown in Fig. 4.6 The introduction of lag load as a feature tends to have a significant relationship between the input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast value. The behaviour of electricity users today and yesterday is similar. The features that are tested with high scores are the Yearly Hours feature, which shows the time position that repeats patterns throughout the year, showing the relationship with the forecast value in a high proportion. These are 2 features that have significantly higher scores than other models. As for other features, the scores are closer to zero and are less relevant to the forecast.

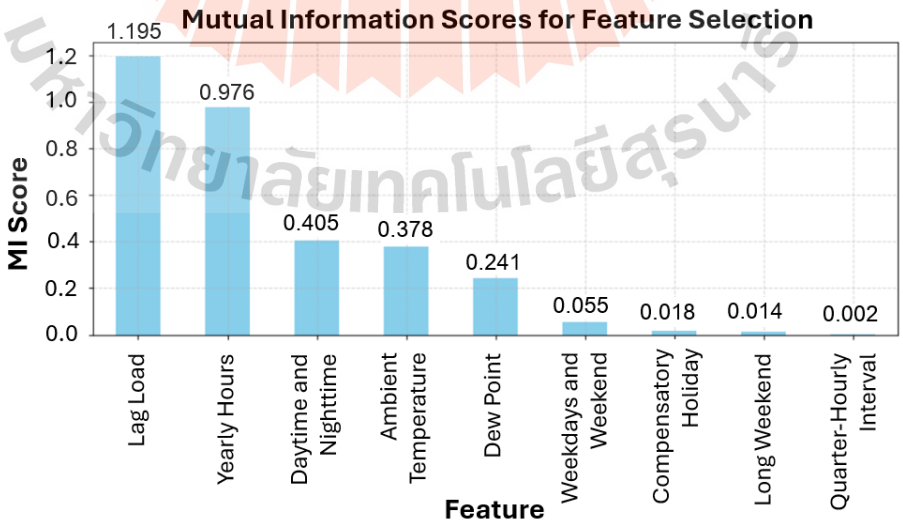


Figure 4.6 Mutual Information Scores of SUT Data

4.10.3 Daily Load Forecasting Performance

Table 4.4 Performance of Each Model on 10 Test Sets

Models	Important dates		Metric Statistic		
	Minimum	Maximum	Metric	Min.	Max.
Moving average	2023-08-01	2023-08-03	MAE (kW)	68.60	675.72
	2023-05-01	2023-08-03	RMSE (kW)	87.39	945.33
	2023-12-27	2023-09-10	MAPE (%)	5.78	78.34
Attention-LSTM	2023-04-16	2023-12-25	MAE (kW)	73.10	385.39
	2023-04-16	2023-12-25	RMSE (kW)	114.97	563.92
	2023-03-01	2023-09-10	MAPE (%)	5.99	35.40
DNN	2023-03-02	2023-12-25	MAE (kW)	97.47	403.80
	2023-03-02	2023-12-25	RMSE (kW)	122.82	615.58
	2023-09-04	2023-09-10	MAPE (%)	10.16	37.38
CNN-LSTM	2023-03-02	2023-05-02	MAE (kW)	68.33	368.25
	2023-03-02	2023-05-02	RMSE (kW)	90.55	556.91
	2023-03-02	2023-09-10	MAPE (%)	7.21	39.63
LSTM	2023-03-02	2023-05-02	MAE (kW)	83.60	362.97
	2023-03-02	2023-05-02	RMSE (kW)	105.69	528.20
	2023-09-03	2023-09-10	MAPE (%)	9.82	33.84
Bi-LSTM	2023-03-02	2023-12-25	MAE (kW)	**46.75**	373.37
	2023-03-02	2023-12-25	RMSE (kW)	**68.35**	566.79
	2023-03-02	2023-09-10	MAPE (%)	**4.82**	38.74
Bi-LSTM (Top 3 Features by SHAP)	2023-12-27	2023-09-11	MAE (kW)	63.52	394.71
	2023-03-03	2023-09-11	RMSE (kW)	90.11	612.98
	2023-03-02	2023-09-10	MAPE (%)	4.92	36.79
Bi-LSTM (Top 3 Features by MI)	2023-03-03	2023-08-03	MAE (kW)	103.28	469.44
	2023-03-03	2023-09-10	RMSE (kW)	141.87	722.92
	2023-03-02	2023-09-10	MAPE (%)	7.51	63.60

One-day-ahead electrical load forecasting at Suranaree University of Technology can demonstrate the performance of each of the six models through three metrics, compiled into statistical metrics that show the highest, lowest, and average values of each metric from ten test data sets. Important days in the forecast, including the days showing the lowest and highest values of each metric, are also included in the test data set for the model. It is shown in Table 4.4. The forecasting results show

that the moving average forecasting method gives the highest value of all 3 indicators from the tested data set. MAE has a value of 675.72 kW, and RMSE has a value of 945.33 kW on 2023-08-03, which is a date in the set of semesters start dates, as shown in Fig. 4.7 And MAPE has a value of 78.34% on 2023-09-10, which is a date in the rainy season set, as shown in Fig. 4.8 The forecasting method with the Bi-LSTM model gives the lowest values of MAE, RMSE, and MAPE indicators from the tested data set, with values of 46.75 kW, 68.35 kW, and 4.82%, respectively, which are a date in the exam set, as shown in Fig. 4.9 The figures show the load forecast for 1 day in advance for 3 consecutive days. Each model is graphed compared to the actual load graph to show the forecast trend.

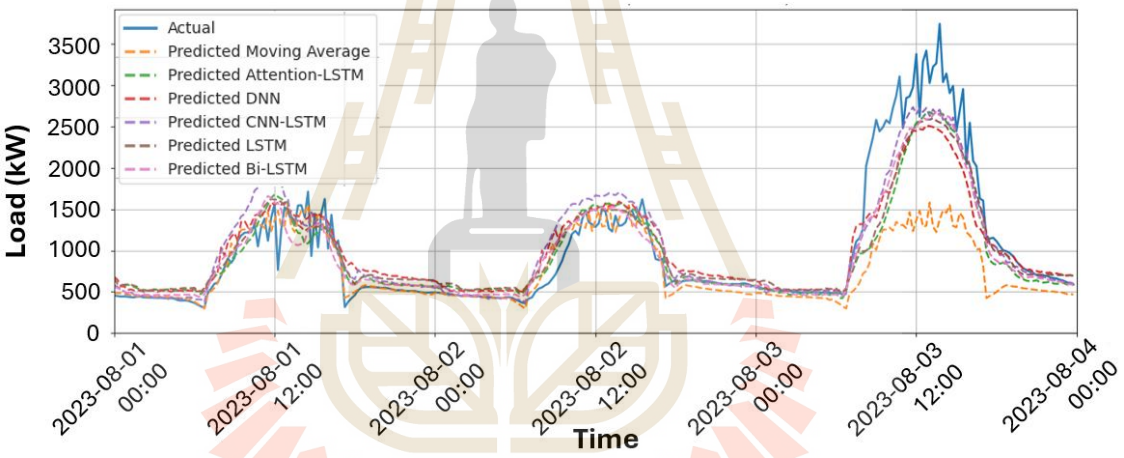


Figure 4.7 The Lowest-Performing Forecast Results in the Initial Semester Test Set

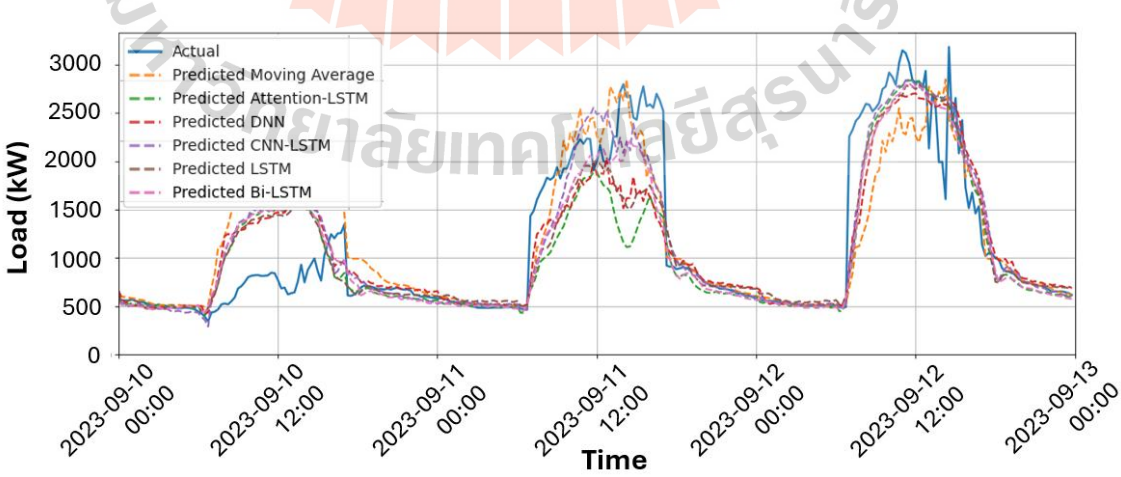


Figure 4.8 The Lowest-Performing Forecast Results in the Rainy Season Test Set

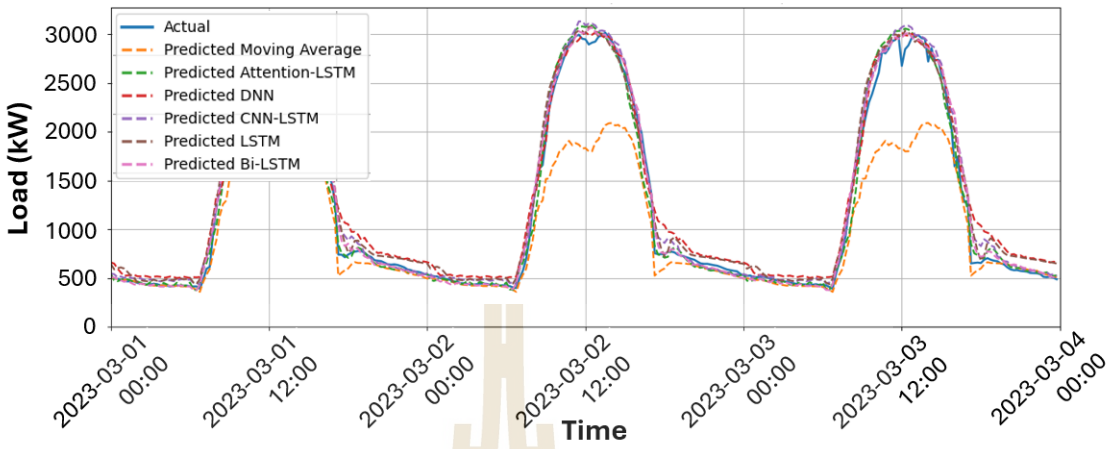


Figure 4.9 The Best-Performing Forecasting Results in the Test Set during the Exam Period

Since the moving average model relies on the average of historical load patterns, they must be similar at the same time point for accurate forecasting. However, data from the beginning of the semester, the rainy season, or the test data used for forecasting are volatile and unstable. This results in higher forecast error rates than all other models when analyzed for the maximum possible forecast error. The Bi-LSTM model has two layers that process historical data and predict future load trends, enabling it to effectively learn from data patterns that fluctuate with weather and time. Furthermore, the test data during exams or similar data have lag load and yearly hours features that significantly affect the relationship with the forecast at that time, accounting for a significant portion of the MI score. This factor occurs both before and after the forecast point. Therefore, the bidirectional learning method demonstrates the highest forecasting performance compared to all other models when analyzed for the maximum possible model accuracy. The Bi-LSTM model exhibits a lower average error (MAE) value, a percentage error compared to the actual value (MAPE), and a lower proportion of large errors (RMSE) than other models. The model can reduce errors at each time point, reduce forecasts with small errors, and

forecast peak load periods more accurately than other models. Therefore, the resulting model is the best-performing model in this context.

The LSTM model showed the lowest values of all three indicators as follows: MAE was 83.60 kW, RMSE was 105.69 kW, and MAPE was 9.82%. The model has a learning characteristic from the past to the present. There is a layer that remembers various values, which is different from the DNN model in that the features fed into the model are not chronologically ordered, which makes the indicator values higher than the LSTM model. MAE was 97.47 kW, RMSE was 122.82 kW, and MAPE was 10.16%. For the hybrid CNN-LSTM model, CNN was used to capture short-term data patterns, resulting in lower errors. The lowest values of all three indicators were as follows: MAE was 68.33 kW, RMSE was 90.55 kW, and MAPE was 7.21%. For the Attention-LSTM model, developed from a simple LSTM structure, it prioritizes the most effective timestep, showing an MAE of 73.10 kW, an RMSE of 114.97 kW, and an MAPE of 5.99%. Both models exhibit higher forecasting accuracy than the simple LSTM model. The moving average method yields the best indicators, with an MAE of 68.60 kW, an RMSE of 87.39 kW, and an MAPE of 5.78%. These are more effective than some of the models mentioned above. However, when examining the worst-day recordings, the model achieves higher indicators than all other models. Furthermore, the average test data also has the lowest overall performance.

In addition, the Bi-LSTM model was investigated to obtain the most three influential features from SHAP: Lag Load, Yearly Hours, Weekdays, and Weekend. The lowest values of the metrics were obtained as follows: MAE was 63.52 kW, RMSE was 90.11 kW, and MAPE was 4.92%. Compared to the same model set using all 9 features, the results show that the model using only the most important features provides acceptable forecast performance, although its performance is slightly reduced compared to using all features. The Lag Load feature has the highest weight in describing the daily electrical load characteristics. The Yearly Hours, Weekdays, and Weekend features still help the model better understand hourly and weekly patterns.

While selecting only 3 important features allows the model to perform well, using all features yields the lowest metric values and the best performance.

The Bi-LSTM model was re-simulated using the top three features identified as having the greatest influence by MI: Lag Load, Yearly Hours, Daytime, and Nighttime. The lowest metrics were MAE at 103.28 kW, RMSE at 141.87 kW, and MAPE at 7.51%. The MI methodology indicates that the Lag Load feature has a strong and consistent relationship with output throughout the year, demonstrating model continuity. The Yearly Hours feature can segment the load by time of year. The daytime and nighttime features can show more detailed time relationships, but they still don't yield results as good as using all nine features. For daily or one-day forecasting, the data structure requires more context than using only three features to allow the model to understand more complex details. Using only three features identifies the most important individual features. Operating alone or without the context of other features fails to recognize true relationships and can lead to higher forecasting errors.

4.10.4 Test Set Load Forecasting Performance

Table 4.5 presents the complete test dataset of the university load data, simulating the results for each of the six models. It also compares simulation cases using the top three most important features from the SHAP method and the MI method, showing the results of the overall test metrics. It was found that the model with the lowest MAE, RMSE, and MAPE metrics was the one selected using the top three most important features from the SHAP method, simulated with the Bi-LSTM model. The MAE was 177.90 kW, the RMSE was 310.29 kW, and the MAPE was 15.59%. These results indicate that the model can efficiently learn the structure of the electricity load data by selecting the most relevant features that truly impact the model. The SHAP method selects features that directly affect the model's output, thus eliminating some less relevant features. Furthermore, the memory of the Bi-LSTM

model is not used for unnecessary features; instead, more weight is given to time-related features, as the model is sensitive to temporal dependency. This results in lower overall metric values compared to other models and methods. The graphs showing the complete test suite for each model are presented in Appendix C.

Table 4.5 Overall Test Set Performance Metrics

Model	Test Set		
	MAE (kW)	RMSE (kW)	MAPE (%)
Attention-LSTM	183.72	323.18	16.68
Bi-LSTM	184.05	321.23	17.53
CNN-LSTM	195.24	332.14	19.21
LSTM	200.78	326.22	20.47
Moving Average	260.56	465.05	20.58
DNN	207.97	330.07	18.23
Bi-LSTM (Top 3 Features of SHAP)	**177.91**	**310.29**	**15.59**
Bi-LSTM (Top 3 Features of MI)	221.87	383.06	19.77

4.10.5 SHAP-Based Feature Importance Analysis

The SHAP tool was imported to describe the importance of features that affect the decision of the Bi-LSTM model, respectively, as shown in Fig. 4.10 The results of the remaining models are shown in the Appendix D. The x-axis shows the SHAP value, with a positive value indicating a higher prediction value. The y-axis sorts the features by cluster importance from highest to lowest, and the color of the dots indicates the actual feature value, with red having a high value and blue having a low value. The top three features of the NN model with the highest importance are as follows: The Lag Load feature is an important variable that has a great impact on the model's decision because it has the highest SHAP value distribution. There are a lot of red dots to the right, meaning that if yesterday's load is higher, it will also push today's

forecast value higher. The Yearly Hours feature is distributed in clusters and has most red dots to the left, meaning that features with high values will lower the forecast value. The Weekdays and Weekend features show that the closer the SHAP value distribution is to zero, the less the feature's impact on the model's decision. In the non-linear features, the SHAP values are widely distributed, with both positive and negative values and a mixture of red and blue dots at the same point. The moving average method is a basic statistical method for forecasting using features Lag 1, Lag 2, and Lag 3 as the electrical load values for the past 1, 2, and 3 days, respectively. The time-related data, namely the Hour and Minute features, are the time data for all 3 days in every hour and minute at the same time of each day. Feature Lag 1 is the feature with the most dispersion and has the most impact on the model's decision.

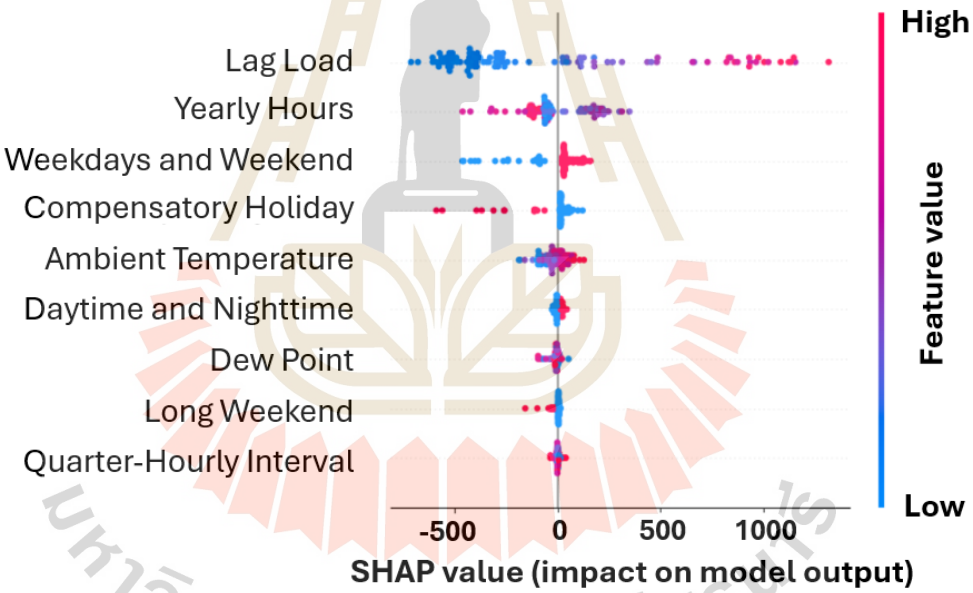


Figure 4.10 SHAP Value of Each Model: Bi-LSTM

4.10.6 Comparative Analysis of Feature Importance Models and Methods

The importance of ranking of features of NN models by SHAP and MI, where model A is LSTM, B is Bi-LSTM, C is CNN-LSTM, D is Attention-LSTM, E is DNN, and F is MI. From Table 4.6, number 1 shows the ranking of the most important features, and the higher the ranking number, the importance will decrease until

number 9, which has the lowest feature importance to the model. The Bi-LSTM model forecasts have the highest accuracy. The ranking of features from the highest to the lowest is shown as follows: Lag Load, Yearly Hours, Weekdays and Weekend, Compensatory Holiday, Ambient Temperature, Daytime and Nighttime, Dew Point, Long Weekend, and Quarter-Hourly Interval, respectively, were compared for all 6 models. The Bi-LSTM and LSTM models are models that learn the time sequence of data. The 4th feature is Compensatory Holiday, which shows the compensatory holidays of the university reflect the overall electricity consumption behavior. There are red and blue dots clearly separated from the blue ones. The model can recognize clear patterns with this data and associate it with the load drop. This differs from the fifth-ranked feature, ambient temperature. Since alternating points in the same location have a non-linear relationship, fluctuations can make it more difficult for models to capture these patterns.

Table 4.6 Feature Importance Ranking of NN Model by SHAP and MI

Feature	Feature importance ranking					
	A	B	C	D	E	F
Lag Load	1	1	1	1	1	1
Yearly Hours	2	2	2	2	2	2
Weekdays and Weekend	3	3	3	3	3	6
Compensatory Holiday	4	4	5	5	5	7
Ambient Temperature	5	5	4	4	4	4
Daytime and Nighttime	9	6	7	8	6	3
Dew Point	6	7	6	6	7	5
Long Weekend	8	8	8	7	8	8
Quarter-Hourly Interval	7	9	9	9	9	9

A CNN-LSTM model, which captures short-term and long-term patterns well, or an Attention-LSTM model that emphasizes time steps, will better understand this feature. Based on feature priorities, CNN-LSTM and Attention-LSTM models prioritize ambient temperature over compensatory holidays. DNN models can easily learn by finding association weights, do not remember event sequences, and process data without considering time sequences. The SHAP value of the Ambient Temperature feature, used to describe load changes, is higher than that of the Compensatory Holiday feature, which relies on time sequences and previous behaviors. The Quarter-Hourly Interval feature, also a later feature, is considered a fundamental feature, as it does not directly correlate with load and is a simple feature for the model to learn. From the simulation results, it was found that the MI method gives more importance to the Daytime and Nighttime, Ambient Temperature, and Dew Point features of each model than the SHAP method because these features have more raw data statistics related to the forecasting target, which reflects the actual behavior of electrical loads without learning the model mechanism and not understanding the relationship between the features like SHAP. When using SHAP, the importance is reduced. For Weekdays and Weekend and Compensatory Holiday features, which have lower MI scores, SHAP gives higher importance to both features. This is because the model learns the effect of these features through the more important Lag Load feature, not just learning the features alone. Using SHAP, one can learn features with complex relationships better.

The third feature of the MI method, Daytime and Nighttime, is shown to be highly correlated with load. However, compared to the SHAP method, this feature is ranked low due to the model's inability to extract patterns well. However, the key features, Lag Load and Yearly Hours, ranked first and second in both methods, showed consistent results, indicating that this feature set can definitely improve accuracy and efficiency. Using load data, the MI method screens out potentially important features related to electrical loads, screening relevant and related data before importing into

the model and isolating truly irrelevant features. The trend in importance of features is evident, eliminating bias from the model for optimal forecasting performance. However, since it is not yet possible to determine how features affect model decisions, SHAP values are used at each data point in real-world situations to explain the relationship between features and model decisions. This ensures that the model is indeed using those features and provides a deeper explanation of model behavior, demonstrating the importance of features that directly influence the model. Furthermore, under different load conditions, the direction of their impact on forecasting can be identified. Regarding the reliability of black-box models, this can be explained.

4.11 Simulation Result and Discussion with ISO New England Data

In electricity load forecasting, understanding the context of the data and the influence of features on the results of various models is crucial. Improving forecasting performance and model description are key issues, as are the repeatability and reliability of the research framework. Therefore, electricity load forecasting was simulated using hourly electricity demand data from January 3, 2003, to December 31, 2014, compiled and published by ISO New England (ISO-NE), an independent agency responsible for regulating and managing electricity systems in the New England region of the United States, as an example to illustrate the analysis of the models and methods proposed above. Using Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models.

4.11.1 Mutual Information Scores of ISO-NE Data

Prioritizing feature data before importing them into the model using the MI tool measures the correlation between features and dip data sets of electricity demand, compiled and published by ISO New England (ISO-NE) of the United States. Both linear and non-linear data are used, showing the ranking scores of each feature. The Lag Load feature has an MI score of 0.977, Yearly Hours 0.487, Ambient

Temperature 0.386, Month 0.171, Weekdays 0.0516, Year 0.023, and Day 0.011, as shown in Fig. 4.11. It can be seen that including lag load as a feature shows a significant correlation between input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast values, with electricity user behavior today and yesterday being very similar. The features with the lowest scores are yearly hours, which show recurring time patterns throughout the year and demonstrate a high correlation with the predicted values. These two features show significantly higher scores than other models. Features with scores close to 0 show decreasing correlation with the predictions in descending order.

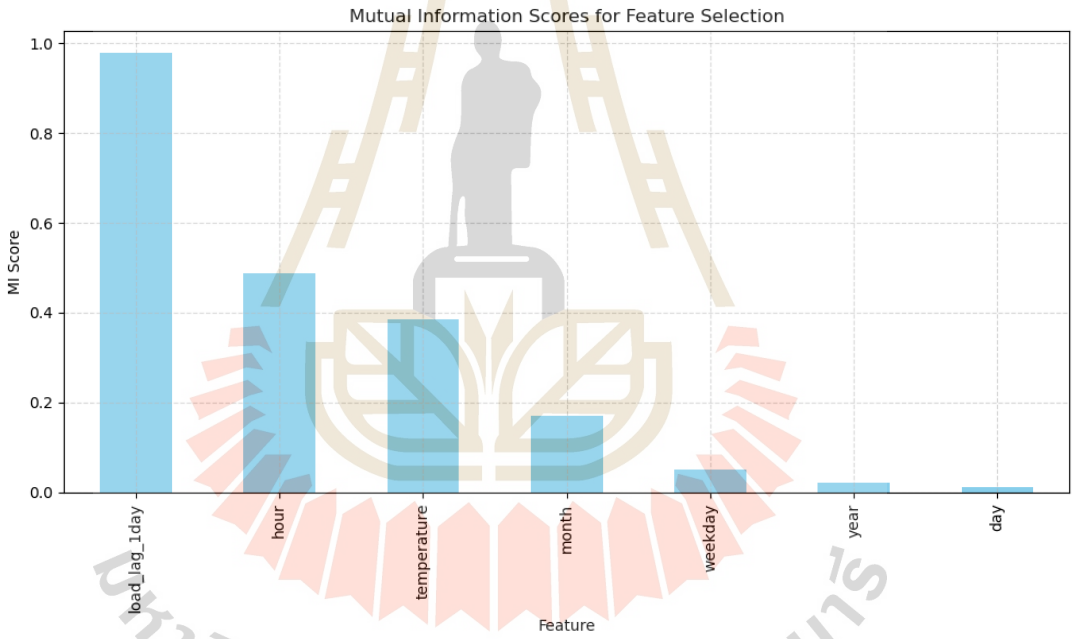


Figure 4.11 Mutual Information Scores of ISO-NE Data

4.11.2 Daily Load Forecasting Performance

The electricity load forecasts compiled and published by ISO New England (ISO-NE) one day in advance can show the performance of each of the six models through three indicators. These are compiled as statistical metrics indicating the lowest values for each model from the tested data. Key forecast dates, including

those showing the lowest values, are also included in the test dataset, as shown in Table 4.7. The forecast results show that the CNN-LSTM method yielded the lowest values for all three indicators from the tested dataset, with MAE at 89.08 kW, RMSE at 118.94 kW, and MAPE at 0.67% on October 9, 2014, making it the most accurate model.

Table 4.7 The best result for each model in daily forecasting

Models	Important dates	Metric Statistic		
	Date of Minimum	MAE (kW)	RMSE (kW)	MAPE (%)
Moving Average	2014-09-25	98.40278	142.3815	0.71
Attention-LSTM	2014-06-10	108.78	137.65	0.74
DNN	2014-07-19	238.51	281.87	1.75
LSTM	2014-10-21	90.73	126.32	0.69
Bi-LSTM	2014-06-05	96.84	123.52	0.70
CNN-LSTM	2014-10-02	**89.08**	**118.94**	**0.67**
CNN-LSTM (Top 3 features SHAP and MI)	2014-10-09	148.25	203.14	1.13

Fig. 4.12 shows a flat graph comparing actual load and other models to illustrate forecast trends for that day. The figure shows three consecutive 1-day load forecasts. The LSTM model yielded the lowest values: MAE 90.73 kW, RMSE 126.32 kW, and MAPE 0.69% on October 21, 2014. The Bi-LSTM model yielded the lowest values: MAE 96.84 kW, RMSE 123.52 kW, and MAPE 0.70% on June 5, 2014. The Moving Average model yielded the lowest values: MAE 98.40 kW, RMSE 142.38 kW, and MAPE 0.71% on date missing. On September 25, 2014, the forecasting method using the Attention-LSTM model yielded the following lowest metric values: MAE at 108.78 kW, RMSE at 137.65 kW, and MAPE at 0.74% on June 10, 2014. The forecasting method using the

DNN model yielded the following lowest metric values: MAE at 238.51 kW, RMSE at 281.87 kW, and MAPE at 1.75% on July 19, 2014. The models display the metric results from lowest to highest values, respectively.

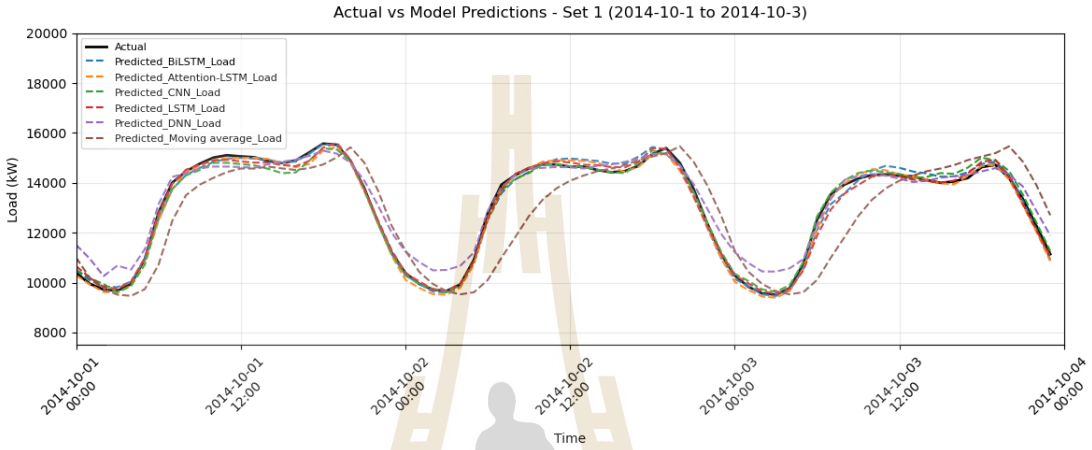


Figure 4.12 Comparison of the best forecasting results of the CNN-LSTM model

The results show that the CNN-LSTM hybrid model is the most accurate for 1-day forecasting. This is because the forecasted data contains recurring daily load changes over short forecast periods. The CNN model analyzes the similarity of these short-term patterns before passing them to the LSTM model to learn longer-term patterns, resulting in the lowest forecast metric. The LSTM model, which is less accurate, primarily learns from time-series memory and performs well on relatively simple datasets. Bi-LSTM models typically learn both past and future patterns, indicating that future pattern learning is not very helpful for this type of dataset. Moving average models, acting as baselines, only learn from historical electricity load data and therefore cannot understand more complex data. Attention-LSTM models can learn from time series datasets that span many steps ahead and are more complex than basic models. However, using them for 1-day forecasting on a fixed-pattern dataset would make the model too complex and unsuitable for the data. The creation of excess parameters leads to worse results compared to other models, and basic DNN

models show the least accurate results because they lack a temporal understanding of forecasting. At each time point, the model does not understand the temporal relationships between features, which is a crucial issue for electricity load forecasting. When there are many parameters used in forecasting, the model is prone to overfitting.

Furthermore, a re-simulation of the most accurate CNN-LSTM model was performed using the top three most influential features identified from SHAP and MI: Lag Load, Yearly Hours, and Ambient Temperature, respectively. Analysis of the data and model, along with statistical methods, indicated that historical load shows the most significant pattern in electricity consumption and has the greatest influence on usage. This is followed by temporal features reflecting important daily patterns, and weather factors tend to reflect electricity usage based on temperature. The lowest values were recorded on October 9, 2014: MAE at 148.25 kW, RMSE at 203.14 kW, and MAPE at 1.13%. The simulation results showed that while the three identified features are highly significant, using only these three results in less accurate forecasting compared to using all features. This is because the model learning process requires diverse datasets and contexts to effectively identify relationships and achieve optimal performance.

4.11.3 Test Set Load Forecasting Performance

Table 4.8 presents the complete test dataset, simulating each of the six models and comparing cases simulating the top three most important features from the SHAP and MI methods with the CNN-LSTM model. The results show the indicators of the entire test set. The model with the lowest MAE is CNN-LSTM, with an MAE value of 333.99 kW. The model with the lowest RMSE is Attention-LSTM, with an RMSE value of 475.15 kW. And the model with the lowest MAPE is CNN-LSTM, with a MAPE value of 2.29%. These results indicate that the CNN-LSTM model is the most stable and accurate, predicting closest to the true values with low MAE (low mean error per data point) and low MAPE (low percentage error across the load range). The Attention-LSTM

model has the lowest RMSE, indicating its ability to control errors occurring during high load periods. This demonstrates the good predictive ability of the entire test dataset during peak load periods. The graphs showing the complete test suite for each model are presented in Appendix C.

Table 4.8 Overall Test Set Performance Metrics of ISO-NE Data

Models	Test Set		
	MAE (kW)	RMSE (kW)	MAPE (%)
Attention-LSTM	334.5389	**475.1453**	2.32
Bi-LSTM	368.1364	534.2110	2.50
CNN-LSTM	**333.9936**	497.3806	**2.29**
LSTM	358.5621	516.5337	2.45
Moving Average	1129.35	1601.82	7.67
DNN	611.4789	824.5546	4.22
CNN-LSTM (Top 3 features SHAP and MI)	625.95	848.3157	4.34

4.11.4 SHAP-Based Feature Importance Analysis

SHAP is used to explain the importance of features affecting the decision-making of Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models, respectively, as shown in Appendix D. The top three most important features in NN models are Lag Load, shown in Fig. 4.13, a key variable significantly impacting model decision-making due to its largest SHAP distribution (shown with red dots close to the right and left, indicating that higher yesterday's load will raise today's forecast, and lower yesterday's load will lower today's forecast); Yearly Hours, which is clustered with purple dots concentrated on the right, suggesting that mid-range values improve forecasts; and Ambient Temperature, where the purple dots are clustered on the left in SHAP distribution, meaning that mid-range values (neither too high nor too low) decrease forecasts. The weekday, month, day, and year features show a SHAP

distribution closer to zero. The impact of features on model decision-making decreases with increasing frequency. The moving average method is a basic method that uses statistical values for forecasting, employing Lag 1, Lag 2, and Lag 3 features as historical electrical load data for the past 1, 2, and 3 days, respectively. The time-related data, the Hour feature, represents hourly data for all three days at the same time each day. The Hour feature is the most widely distributed and has the greatest impact on model decision-making.

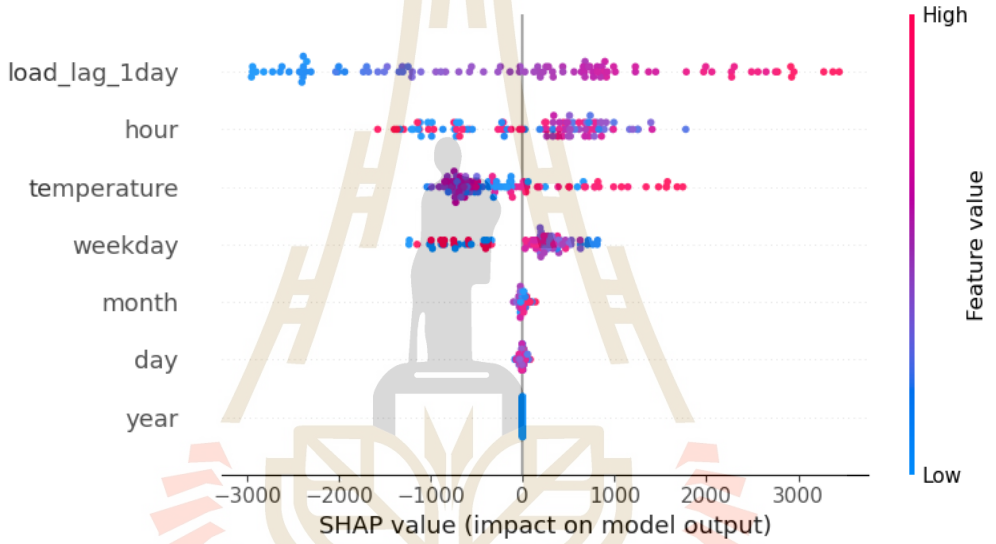


Figure 4.13 SHAP Value of Each Model of ISO-NE Data: CNN-LSTM

4.11.5 Comparative Analysis of Feature Importance Models and Methods

Table 4.9 shows the priority of features from highest to lowest, with increasing priority as the number increases, down to feature number 9, which has the lowest importance to the model. CNN-LSTM models show the highest forecasting accuracy. Based on SHAP analysis, the priority of features from highest to lowest is as follows: Lag Load, Yearly Hours, Ambient Temperature, Weekday, Month, Day, and Year. A comparison of the six models is shown. LSTM and Attention-LSTM models have similar priorities. Bi-LSTM models have a similar priority, with features 2 (temperature) and 3 (hourly) swapped. Because the models learn in two directions within the same

sequence, there is temporal continuity, and the temperature feature exhibits this continuity, giving it a higher priority in Bi-LSTM. The hourly feature, however, does not translate according to time but simply indicates the position of a 24-hour cycle. DNN models have a similar priority, with features 5 (Day) and 6 (Month) swapped, as temporal continuity is a factor. The model does not recognize the time sequence and directly affects the load. However, the Day feature clearly impacts electricity user behavior, making it easy for the model to grasp the relationship, thus ranking higher in DNN models.

Table 4.9 Feature Importance Ranking of ISO-NE Data

Feature	Feature importance ranking of NN model by SHAP and MI					
	LSTM	Bi-LSTM	CNN-LSTM	Attention-LSTM	DNN	MI
Lag Load	1	1	1	1	1	1
Yearly Hours	2	3	2	2	2	2
Ambient Temperature	3	2	3	3	3	3
Weekday	4	4	4	4	4	5
Month	5	5	5	5	6	4
Day	6	6	6	6	5	7
Year	7	7	7	7	7	6

Based on the MI analysis, the feature priority is similar to that of the CNN-LSTM model by SHAP, but with some positional changes. Feature position 4 represents the Month feature, and feature position 5 represents the Weekday feature. Feature positions 6 represent the Year feature, and 7 represents the Day feature. The MI method views the overall statistical relationship of the data, prioritizing features that result in the lowest forecasting error. The Month feature highlights different seasons, and since the MI method considers the overall year and therefore has lower

error margins, it is more important than the Weekday feature, which recurs over short periods. Similarly, the Year and Day features are given higher priority by the MI method because they can better separate load groups over longer periods.

4.12 Summary of Results

This research proposes the development of forecasting techniques using deep learning. The accuracy of the models was compared to underlying neural network (NN) models. Bi-LSTM and CNN-LSTM models showed the highest accuracy and performance for forecasting daily electricity consumption. Data collected, after cleaning and feature generation, was transformed into features ready for input into forecasting models. Electricity consumption forecasts were most consistent with actual values. Overall, the Bi-LSTM model with feature selection using the SHAP method, Attention-LSTM, and CNN-LSTM provided the most accurate forecasts and also yielded the lowest average metrics among all tested datasets, indicating the models' potential for forecasting in various contexts.

Furthermore, the model reliability investigation indicates that transparent feature analysis and model description using MI can explain the relationships between raw data and electricity loads and filter static data for inspection before inputting the model. This reduces the number of features required for model input and reduces training time. SHAP provides in-depth model descriptions, demonstrating the actual use of the model's features, understanding complex features, and increasing model reliability because all features have underlying relationships that can be demonstrated. Using this combined approach, comprehensive coverage is achieved. It provides accurate forecasting results and can also provide causal explanations.

CHAPTER V
CONCLUSION

5.1 Conclusion

This study proposes a method for developing a one-day advance electricity load forecasting model using data from Suranaree University of Technology. It employs deep learning techniques, including Day-Aware Data Arrangement (DGA), to compare deep learning models such as LSTM, GRU, Bi-LSTM, and Bi-GRU. The results indicate that LSTM and Bi-LSTM tend to provide the most accurate forecasts, with GRU showing the fastest simulation time in initial testing. Subsequently, selected models—Deep Learning, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM—were further selected for comparison with baseline models to improve accuracy and reliability. This was achieved through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis, optimized through data collection and feature engineering tailored to the context of the university's integrated solar power generation capacity. Using three accuracy indices—Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE)—the model from the NN group, the Bi-LSTM model using the SHAP 3 feature selection method for the top 3 most important features, provided the best overall forecasting performance from all test datasets. The Bi-LSTM model was also the most accurate, yielding the lowest metric values for daily forecasts compared to all other models. Feature selection that aligns with the model's learning mechanism has a greater impact on performance than considering statistical relationships alone. However, experimental results also indicate that using only the top three features does not always yield the best results. In highly complex datasets, such as those with holidays, seasons, and weather influences, using multiple features allows the model to learn the data context more comprehensively. Another dataset used for simulation, the ISO New England dataset, showed that the Attention-LSTM model performed best in forecasting electricity load across all test datasets. With the

lowest RMSE and MAE/MAPE values, CNN-LSTM is the most suitable model for this type of dataset, exhibiting the best overall forecasting performance. The CNN-LSTM model also shows the highest accuracy in daily pre-load forecasting. Time-sequence models, particularly Bi-LSTM and CNN-LSTM, significantly outperform time-structure insensitive models, reflecting the fact that the ability to learn past and future relationships in time-sequence data is a crucial factor in electricity load forecasting accuracy.

Analysis of the features that have the greatest impact and influence on electricity load forecasting using both MI statistical analysis and SHAP principles revealed that the lag load feature, or the value of electricity load over the past 1 day, is the feature most correlated with the forecasting target and has the greatest impact on the model's decision-making. This characteristic was consistent across both datasets. MI, as a statistical measure, can capture nonlinear relationships and is easy to use, but it's unclear how the feature is actually utilized by the model. It's suitable for initial feature screening and as a statistical indicator of how much a feature shares information with the target variable, independent of any model. In contrast, SHAP analysis can show the relationship between data and the model, allowing for the selection of necessary features and reducing forecasting errors. It's suitable for datasets that show relatively consistent electricity load behavior, clear time-repeating patterns, and are driven by only a few key features, leading to more accurate forecasts. It's ideal for in-depth feature selection. However, the priority of features obtained from MI and SHAP differs due to their different objectives.

The performance of electricity load forecasting does not depend solely on the model or feature selection method but also on the consistency between the data structure, model, and shared feature set. The findings can be applied to the design of electricity load forecasting systems in the context of smart energy systems and provide guidance for selecting appropriate models and features for future time-scheduling forecasting tasks.

5.2 Suggestions

Based on the findings of this study, several recommendations can be proposed to enhance the applicability and robustness of short-term electricity load forecasting models in future research. First, the incorporation of additional explanatory variables, such as electricity price signals, socioeconomic factors, and distributed energy resource penetration, may further improve forecasting accuracy and model generalization under more diverse operating conditions.

Second, although this study demonstrates the effectiveness of advanced recurrent neural network architectures combined with feature selection and interpretability techniques, future work could explore hybrid and ensemble learning strategies to leverage the complementary strengths of different model structures. The integration of probabilistic forecasting approaches is also recommended to provide uncertainty information that can support more informed decision-making in power system operation and planning.

Third, the use of advanced model interpretability techniques could be further extended to investigate dynamic feature importance across different time periods and operating scenarios. This would provide deeper insights into how electricity consumption patterns evolve over time and under varying conditions, particularly during extreme events or atypical load behaviors.

Finally, future studies may consider applying the proposed forecasting framework to larger-scale power systems or multi-region datasets with different load characteristics to further validate its robustness and generalizability. Such extensions would contribute to the development of reliable, interpretable, and practical electricity load forecasting systems for real-world applications.

REFERENCES

Abumohsen, M., Owda, A. Y., & Owda, M. (2023). Electrical load forecasting using LSTM, GRU, and RNN algorithms. Energies, 16(5), 2283. https://doi.org/10.3390/en16052283

Atef, S., & Eltawil, A. B. (2020). Assessment of stacked unidirectional and bidirectional long short-term memory networks for electricity load forecasting. Electric Power Systems Research, 189, 106489. https://doi.org/10.1016/j.epsr.2020.106489

Cordeiro-Costas, M., Villanueva, D., Eguía-Oller, P., Martínez-Comesaña, M., & Ramos, S. (2023). Load forecasting with machine learning and deep learning methods. Applied Sciences, 13(13), 7933. https://doi.org/10.3390/app13137933

Fan, S., & Hyndman, R. J. (2012). Short-term load forecasting based on a semi-parametric additive model. IEEE Transactions on Power Systems, 27(1), 134–141. https://doi.org/10.1109/TPWRS.2011.2162082

Habbak, H., Mahmoud, M., Metwally, K., Fouda, M. M., & Ibrahem, M. I. (2023). Load forecasting techniques and their applications in smart grids. Energies, 16(3), 1480. https://doi.org/10.3390/en16031480

Jang, J., Jeong, W., Kim, S., Lee, B., Lee, M., & Moon, J. (2023). RAID: Robust and interpretable daily peak load forecasting via multiple deep neural networks

and Shapley values. Sustainability, 15(8), 6951. https://doi.org/10.3390/su15086951

Koprinska, I., Rana, M., & Agelidis, V. G. (2015). Correlation and instance based feature selection for electricity load forecasting. Knowledge-Based Systems, 82, 29–40. https://doi.org/10.1016/j.knosys.2015.02.017

Lee, Y.-G., Oh, J.-Y., Kim, D., & Kim, G. (2023). SHAP value-based feature importance analysis for short-term load forecasting. Applied Science and Convergence Technology, 18, 579–588. https://doi.org/10.1007/s42835-022-01161-9

Liu, X., Yang, Z., Guo, Y., Li, Z., & Xu, X. (2024). A novel correlation feature self-assigned Kolmogorov-Arnold Networks for multi-energy load forecasting in integrated energy systems. Energy Conversion and Management, 300, 119388.

Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy usage forecasting model based on long short-term memory (LSTM) and explainable artificial intelligence (XAI). Information, 14(5), 265. https://doi.org/10.3390/info14050265

Mpawenimana, I., Pegatoquet, A., Roy, V., Rodriguez, L., & Belleudy, C. (2020). A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing. 2020 IEEE Sensors Applications Symposium (SAS), 1–6. https://doi.org/10.1109/SAS48726.2020.9220021

Neubauer, A., Brandt, S., & Kriegel, M. (2025). Explainable multi-step heating load forecasting: Using SHAP values and temporal attention mechanisms for enhanced interpretability. Energy and AI, 100480. https://doi.org/10.1016/j.egyai.2025.100480

Panapongpakorn, T., & Banjerdpongchai, D. (2019). Short-term load forecast for energy management system using neural networks with mutual information method of input selection. 2019 SICE International Symposium on Control Systems (SICE ISCS), 13–18. https://doi.org/10.23919/SICEISCS.2019.8758714

Pavlatos, C., Makris, E., Fotis, G., Vita, V., & Mladenov, V. (2023). Enhancing electrical load prediction using a bidirectional LSTM neural network. Electronics, 12(22), 4652. https://doi.org/10.3390/electronics12224652

Pirbazari, A. M., Chakravorty, A., & Rong, C. (2019). Evaluating feature selection methods for short-term load forecasting. 2019 IEEE International Conference on Big Data and Smart Computing (BigComp). https://doi.org/10.1109/BIGCOMP.2019.8679188

Prakash, S., Jalal, A. S., & Pathak, P. (2023). Forecasting COVID-19 pandemic using Prophet, LSTM, hybrid GRU-LSTM, CNN-LSTM, Bi-LSTM and Stacked-LSTM for India. 2023 6th International Conference on Information Systems and Computer Networks (ISCON), 1–6. https://doi.org/10.1109/ISCON57294.2023.10112065

Shi, H., Xu, M., & Li, R. (2018). Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Transactions on Smart Grid, 9(5), 5271–5280. https://doi.org/10.1109/TSG.2017.2686012

Sias, Q. A., Gantassi, R., & Choi, Y. (2024). Multivariate bidirectional gate recurrent unit for improving accuracy of energy prediction. ICT Express. https://doi.org/10.1016/j.icte.2024.10.002

Ullah, K., Khan, M. A., Hassan, M. A., Ullah, A., & Lee, M. Y. (2024). Short-term load forecasting: A comprehensive review and simulation study with CNN–LSTM hybrids approach. IEEE Access, 12, 111858–111881. https://doi.org/10.1109/ACCESS.2024.3440631

Vanitha, R., & Swaroopan, N. M. J. (2023). Contrast of stacked GRU-RNN and bidirectional LSTM network based process for forecasting renewable energy for smart grid. 2023 International Conference on RMKMATE, Chennai, India. https://doi.org/10.1109/RMKMATE59243.2023.10369285

Van Zyl, A., Heyns, T., & Bekker, J. (2024). Exploring the effectiveness of SHAP for feature selection in national demand forecasting. Energy Reports, 10, 152–162. https://doi.org/10.1016/j.egyr.2024.01.012

van Zyl, C., Ye, X., & Naidoo, R. (2023). Harnessing explainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP. Applied Energy, 344, 122079. https://doi.org/10.1016/j.apenergy.2023.122079

Wang, J., Yu, B., Chen, X., Dai, G., Liu, W., He, N., Zhu, P., Yin, Z., & Pan, Z. (2025). An interpretable short-term electrical load forecasting model based on SHapley Additive exPlanations—A case study in Haidian, Beijing. Electric Power Systems Research, 226, 111769. https://doi.org/10.1016/j.epsr.2025.111769

Wu, P., Huang, Z., Pian, Y., Xu, L., Li, J., & Chen, K. (2020). A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting. Journal of Advanced Transportation, 2020, 1–12. https://doi.org/10.1155/2020/8863724

มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX

มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX A

Performance of LSTM model on 12 test sets

Table A.1 Performance of LSTM Model on 12 Test Sets

Set	Data types	Date of data set	Performance of LSTM models		
			MAPE (%)	MAE (kW)	RMSE (kW)
1	Long holiday	2023-01-02	15.76	102.81	156.36
		2023-01-03	12.27	134.20	228.64
		2023-01-04	11.34	97.46	126.41
2	Weekdays during the school term	2023-02-01	24.72	397.40	765.97
		2023-02-02	12.36	180.62	283.29
		2023-02-03	24.01	280.92	424.00
3	Exam period	2023-03-01	6.34	90.21	157.10
		2023-03-02	**5.93**	83.05	145.60
		2023-03-03	10.59	126.19	192.18
4	Summer season	16.49	135.60	199.88	16.49
		8.49	68.27	102.64	8.49
		9.31	111.19	188.19	9.31
5	Weekdays during the school term	8.24	90.70	131.63	8.24
		20.35	298.40	452.52	20.35
		18.12	288.76	439.26	18.12
6	The start of the semester	2023-08-01	13.39	110.15	173.10
		2023-08-02	6.78	**59.18**	**93.66**
		2023-08-03	15.00	313.06	486.03

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table A.1 Performance of LSTM Model on 12 Test Sets (continued)

Set	Data types	Date of data set	Performance of LSTM models		
			MAPE (%)	MAPE (%)	MAPE (%)
7	Compensation holiday period	2023-09-02	13.10	141.77	226.18
		2023-09-03	11.15	104.84	150.12
		2023-09-04	11.09	236.59	401.60
8	Rainy season	2023-09-10	38.89	277.84	415.75
		2023-09-11	15.71	306.48	490.29
		2023-09-12	12.21	207.54	344.20
9	Weekdays and weekends	2023-10-01	9.72	122.79	233.52
		2023-10-02	10.31	187.96	314.12
		2023-10-03	12.33	173.16	256.57
10	Exam period	2023-10-16	10.99	200.85	330.97
		2023-10-17	8.72	148.79	230.52
		2023-10-18	9.05	149.75	247.86
11	Term break	2023-11-01	13.50	201.88	318.32
		2023-11-02	12.40	191.58	301.04
		2023-11-03	19.65	229.27	350.74
12	Winter season	2023-12-25	26.53	404.89	625.17
		2023-12-26	12.48	187.36	325.77
		2023-12-27	19.21	354.07	564.38

มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX B

Implementation and Experimental Setup

B.1 Bi-LSTM model forecasting code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import TimeSeriesSplit
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import mean_absolute_error, mean_squared_error


# ---------- Helper Functions ---------- #
def mean_absolute_percentage_error_manual(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
def create_bilstm_model(input_shape, lstm_units=100, dropout_rate=0.2,
learning_rate=0.001):
    model = Sequential()
    model.add(Bidirectional(LSTM(units=lstm_units), input_shape=input_shape))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1))
    optimizer = Adam(learning_rate=learning_rate)
```

```
    model.compile(optimizer=optimizer, loss='mse')
```

B.1 Bi-LSTM model forecasting code (continued)

```
    return model
def create_sequences(X, y, timesteps):
    Xs, ys = [], []
    for i in range(len(X) - timesteps):
        Xs.append(X[i:i+timesteps])
        ys.append(y[i+timesteps])
    return np.array(Xs), np.array(ys)


# ------------------ Load Dataset ------------------ #
file_train_val = r"D:\New folder\Load forcasting 02\Data\trainNvalidation.xlsx"
file_test = r"D:\New folder\Load forcasting 02\Data\test.xlsx"
df_train_val = pd.read_excel(file_train_val)
df_test = pd.read_excel(file_test)


# ------------------ Prepare Data ------------------ #
selected_features = [
    'Yearly Hours', 'Quarter-Hourly Interval', 'Weekdays and Weekend',
    'Daytime and Nighttime', 'Compensatory Holiday', 'Long Weekend',
    'Lag_1Day_Smoothed_Load', 'Ambient Temperature', 'Relative Humidity']
features = df_train_val[selected_features].values
target = df_train_val['Pure Load '].values.reshape(-1, 1)
test_features = df_test[selected_features].values
test_target = df_test['Pure Load '].values.reshape(-1, 1)
test_time = df_test.iloc[:, 0]
```

B.1 Bi-LSTM model forecasting code (continued)

```
# Normalize
scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()
X_scaled = scaler_X.fit_transform(features)
y_scaled = scaler_y.fit_transform(target)
timesteps = 8
X_seq, y_seq = create_sequences(X_scaled, y_scaled, timesteps)

# ---------- K-Fold with TimeSeriesSplit ---------- #
tscv = TimeSeriesSplit(n_splits=5)
val_metrics = []
fold = 1
for train_idx, val_idx in tscv.split(X_seq):
    X_train, X_val = X_seq[train_idx], X_seq[val_idx]
    y_train, y_val = y_seq[train_idx], y_seq[val_idx]
    model = create_bilstm_model((timesteps, X_train.shape[2]))
    early_stop = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
    model.fit(X_train, y_train, epochs=50, batch_size=32,
              validation_data=(X_val, y_val),
              callbacks=[early_stop], verbose=0)
    y_val_pred_scaled = model.predict(X_val)
    y_val_pred = scaler_y.inverse_transform(y_val_pred_scaled)
    y_val_actual = scaler_y.inverse_transform(y_val)
    mae_val = mean_absolute_error(y_val_actual, y_val_pred)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    rmse_val = np.sqrt(mean_squared_error(y_val_actual, y_val_pred))
```

B.1 Bi-LSTM model forecasting code (continued)

```
    mape_val = mean_absolute_percentage_error_manual(y_val_actual, y_val_pred)
    val_metrics.append((mae_val, rmse_val, mape_val))
    print(f"\nFold {fold} Validation Metrics:")
    print(f"MAE: {mae_val:.4f} kW, RMSE: {rmse_val:.4f} kW, MAPE: {mape_val:.2f}%")
    fold += 1


# ------------------ Evaluate on Test Set ------------------ #
X_test_scaled = scaler_X.transform(test_features)
y_test_scaled = scaler_y.transform(test_target)
X_test_seq, y_test_seq = create_sequences(X_test_scaled, y_test_scaled, timesteps)
test_time = test_time[timesteps:]
y_test_pred_scaled = model.predict(X_test_seq)
y_test_pred = scaler_y.inverse_transform(y_test_pred_scaled)
y_test_actual = scaler_y.inverse_transform(y_test_seq)
test_df = pd.DataFrame({
    "Time": test_time.reset_index(drop=True),
    "Actual": y_test_actual.flatten(),
    "Predicted": y_test_pred.flatten()
})
test_df['Date'] = pd.to_datetime(test_df['Time']).dt.date
print("\nTest Set Metrics by Day:")
for date in test_df['Date'].unique():
    daily = test_df[test_df['Date'] == date]
    mae = mean_absolute_error(daily['Actual'], daily['Predicted'])
    rmse = np.sqrt(mean_squared_error(daily['Actual'], daily['Predicted']))
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    mape = mean_absolute_percentage_error_manual(daily['Actual'], daily['Predicted'])
```

B.1 Bi-LSTM model forecasting code (continued)

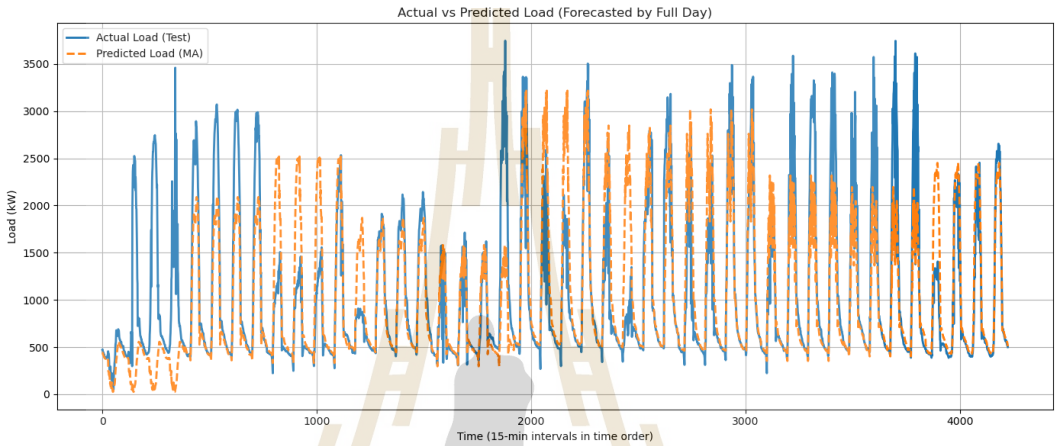
```
    print(f"{date} -> MAE: {mae:.2f} kW, RMSE: {rmse:.2f} kW, MAPE: {mape:.2f}%")

# ------------------ Plot Results ------------------ #
plt.figure(figsize=(14, 6))
plt.plot(y_val_actual, label='Actual Load (Validation)', linestyle='-', linewidth=2)
plt.plot(y_val_pred, label='Predicted Load (Validation)', linestyle='--', linewidth=2)
plt.title('Actual vs Predicted Load (Validation Set - Last Fold)')
plt.xlabel('Time (15-min intervals)')
plt.ylabel('Load (kW)')
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
plt.figure(figsize=(14, 6))
plt.plot(test_df['Actual'].values, label='Actual Load (Test)', linestyle='-', linewidth=2)
plt.plot(test_df['Predicted'].values, label='Predicted Load (Test)', linestyle='--',
linewidth=2)
plt.title('Actual vs Predicted Load (Test Set)')
plt.xlabel('Time (15-min intervals)')
plt.ylabel('Load (kW)')
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

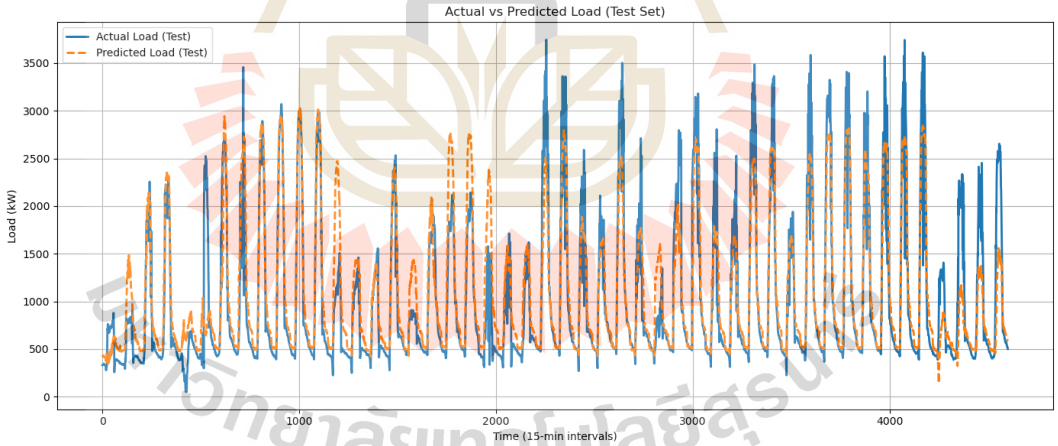
มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX C

Test Set Load Forecasting Performance

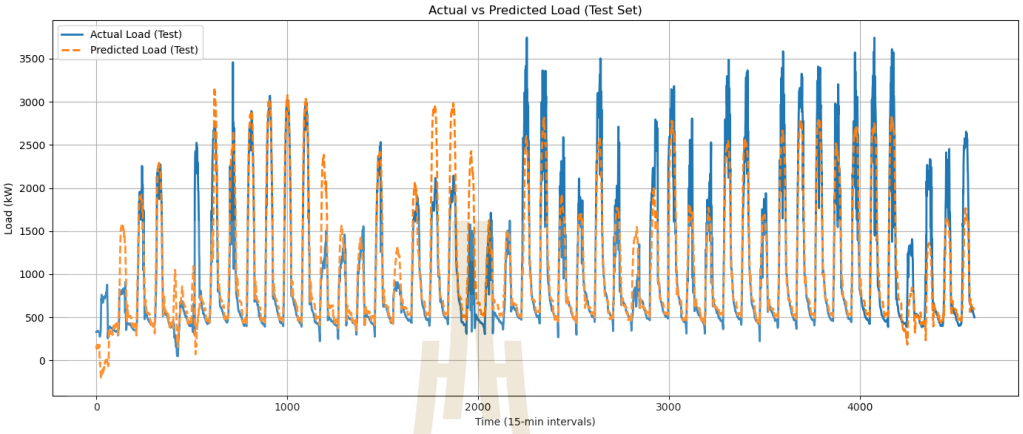


(a)

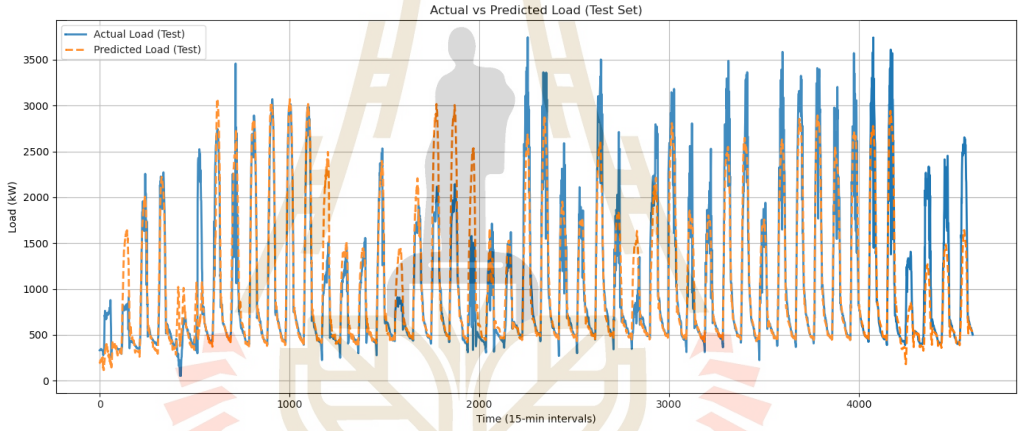


(b)

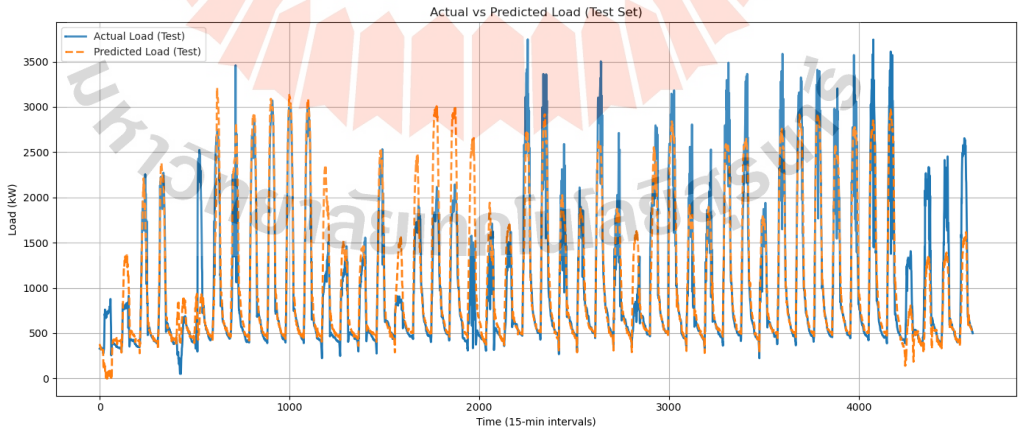
Figure C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM



(c)

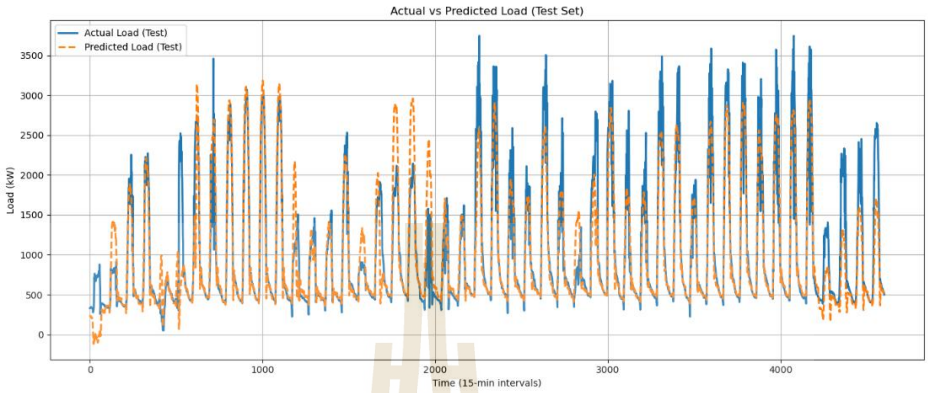


(d)



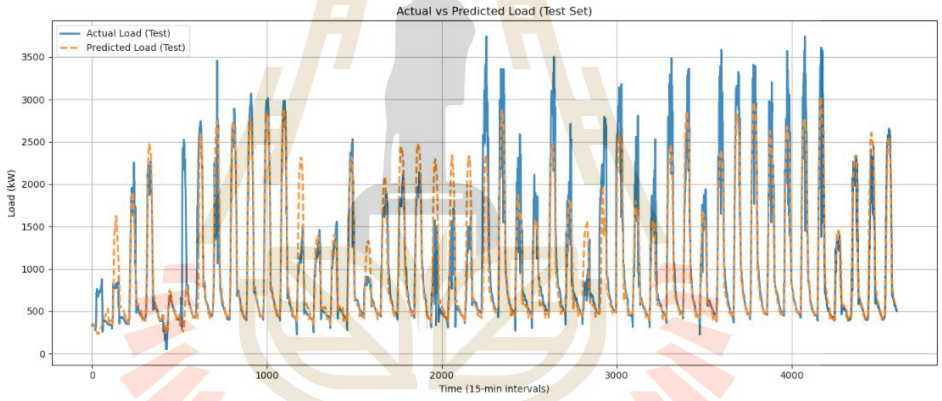
(e)

Figure C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

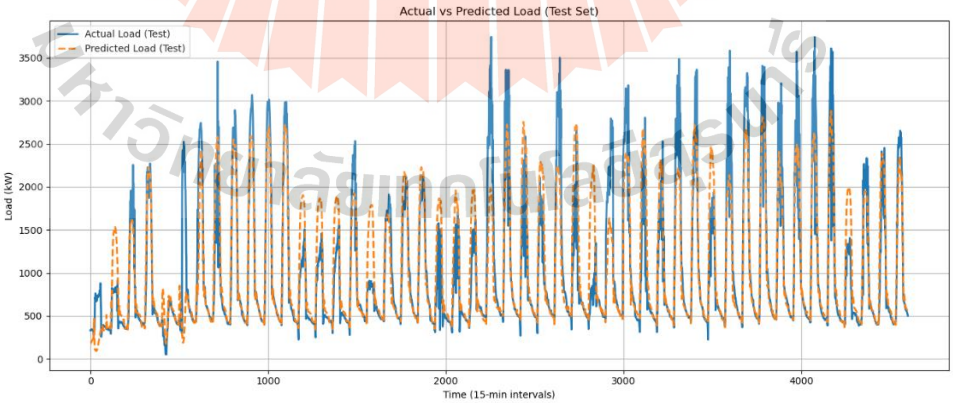


(f)

Figure C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

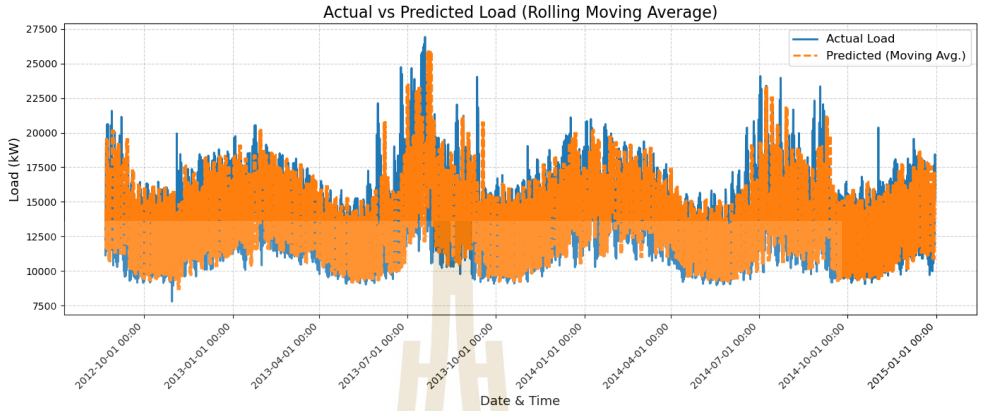


(a)

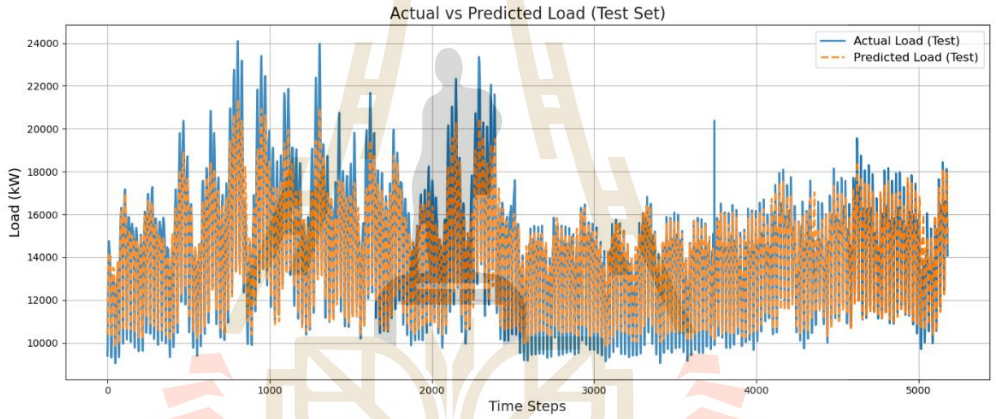


(b)

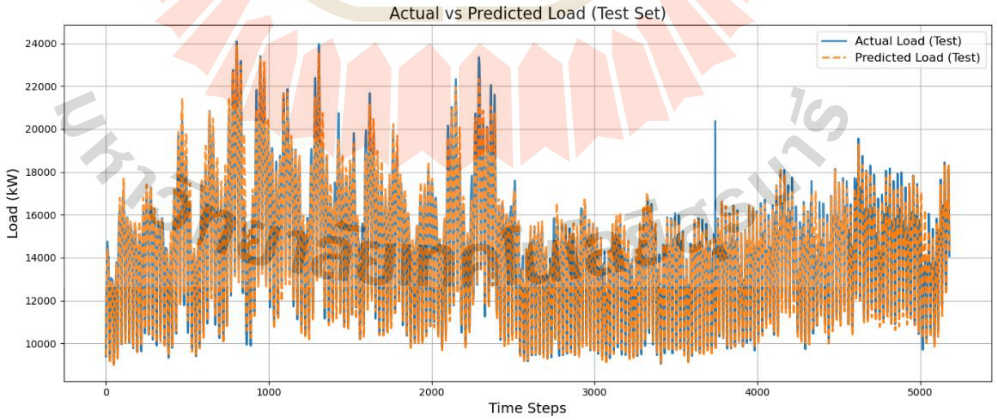
Figure C.2 Performance evaluation graph of the model's test suite: a) Bi-LSTM (Top 3 Features of SHAP) and b) Bi-LSTM (Top 3 Features of MI)



(a)



(b)

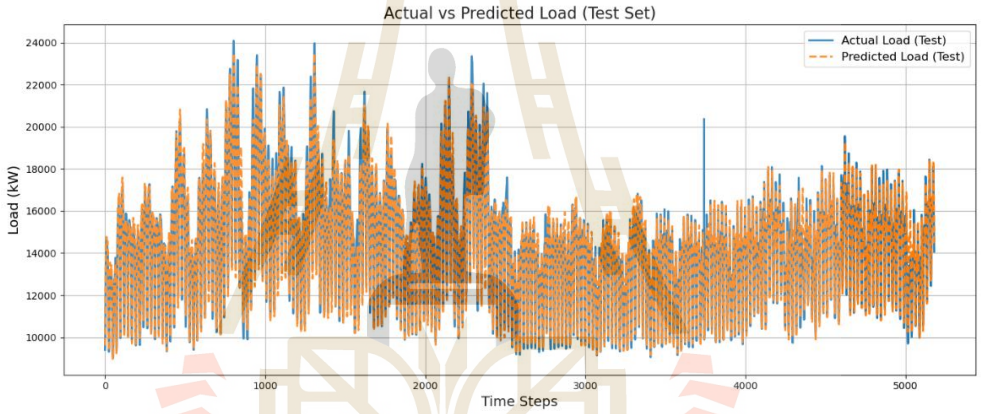


(c)

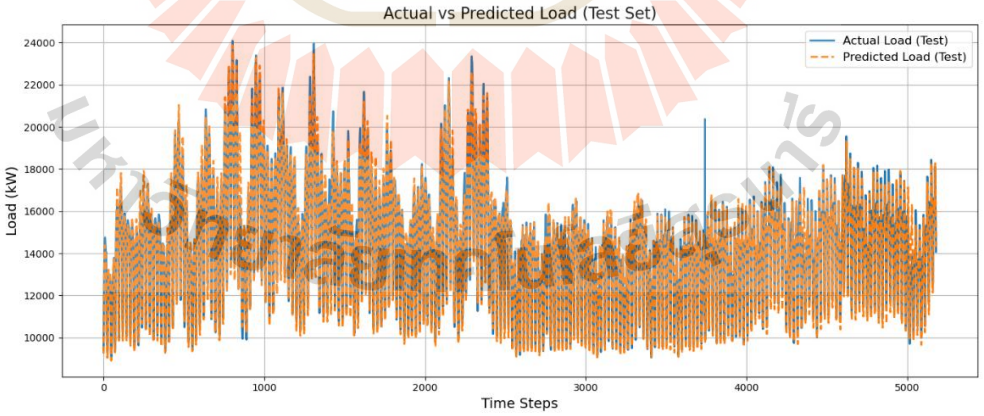
Figure C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features



(d)

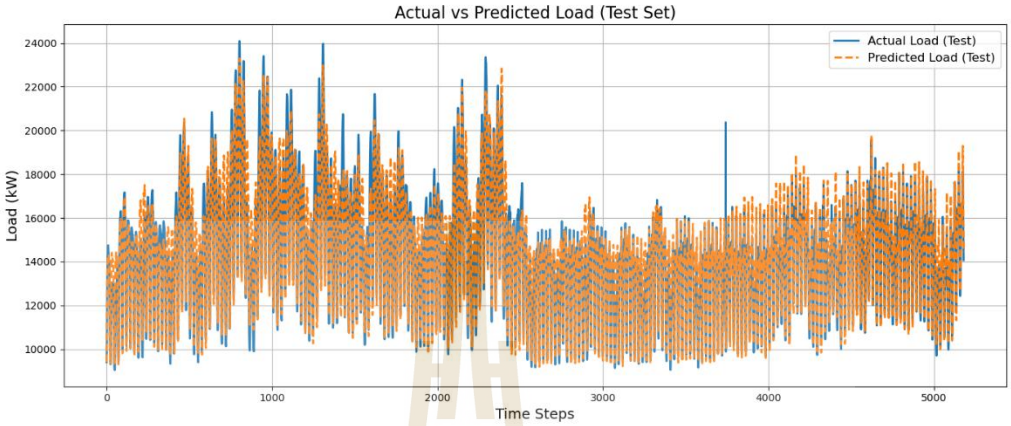


(e)



(f)

Figure C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features (continued)



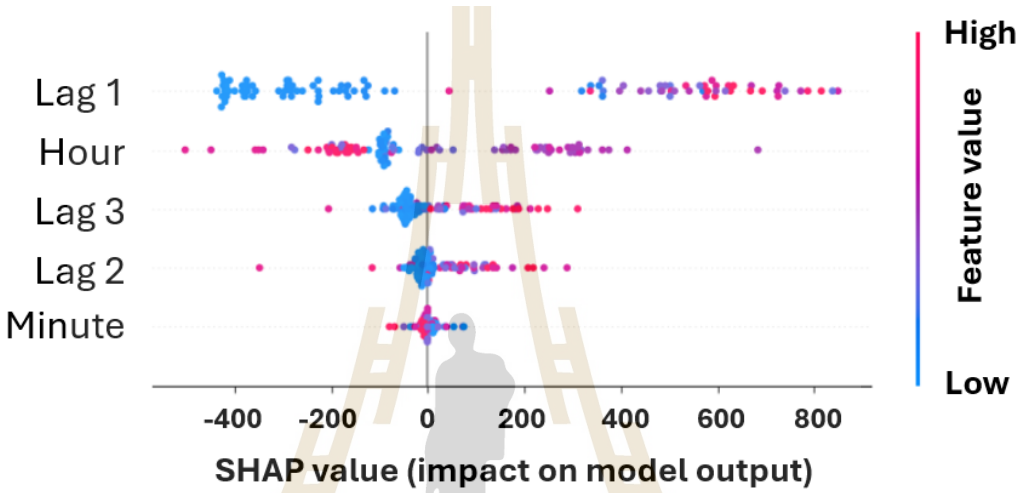
(g)

Figure C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features

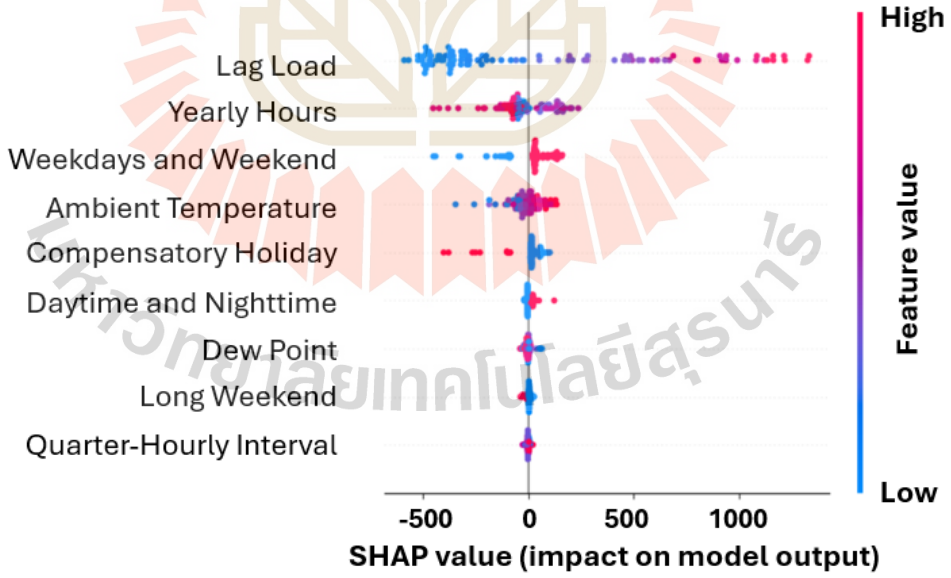
มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX D

SHAP-Based Feature Importance Analysis

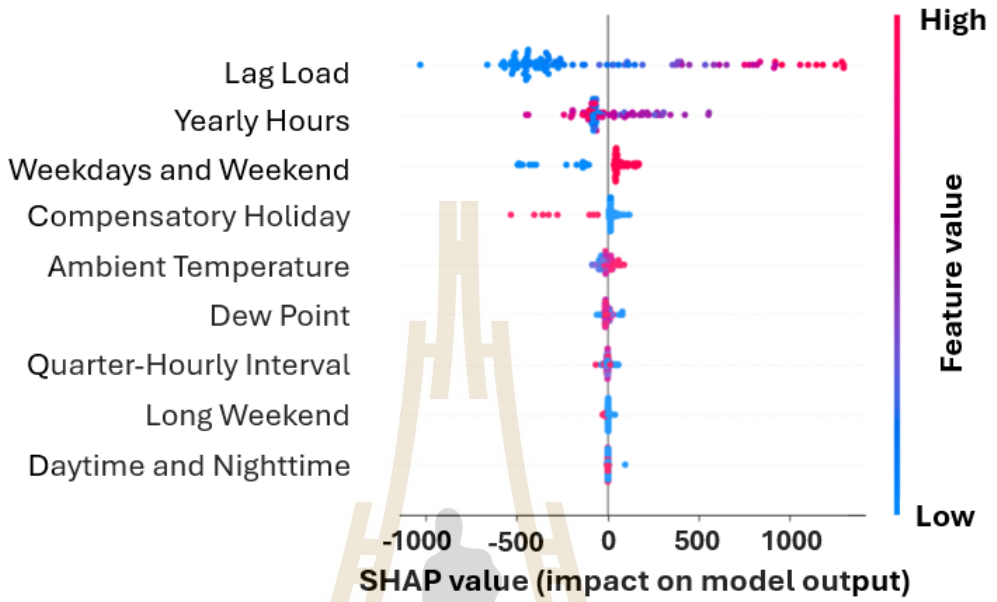


(a)

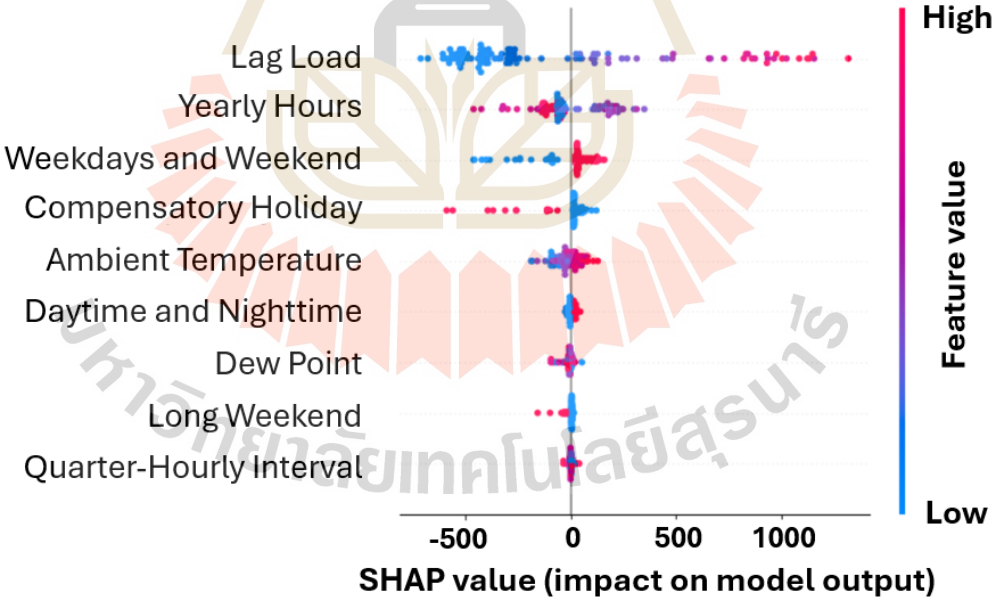


(b)

Figure D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM

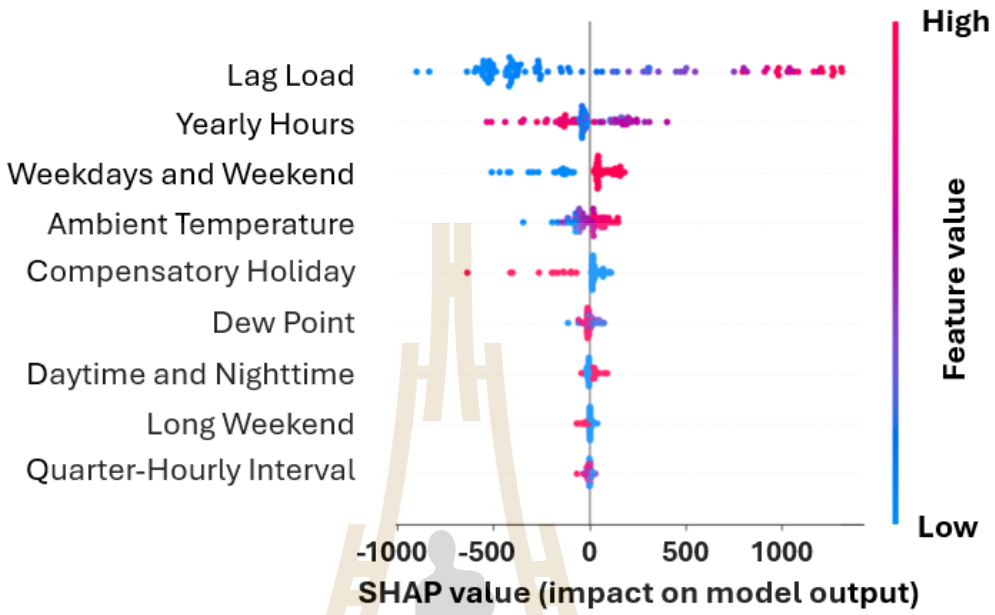


(c)

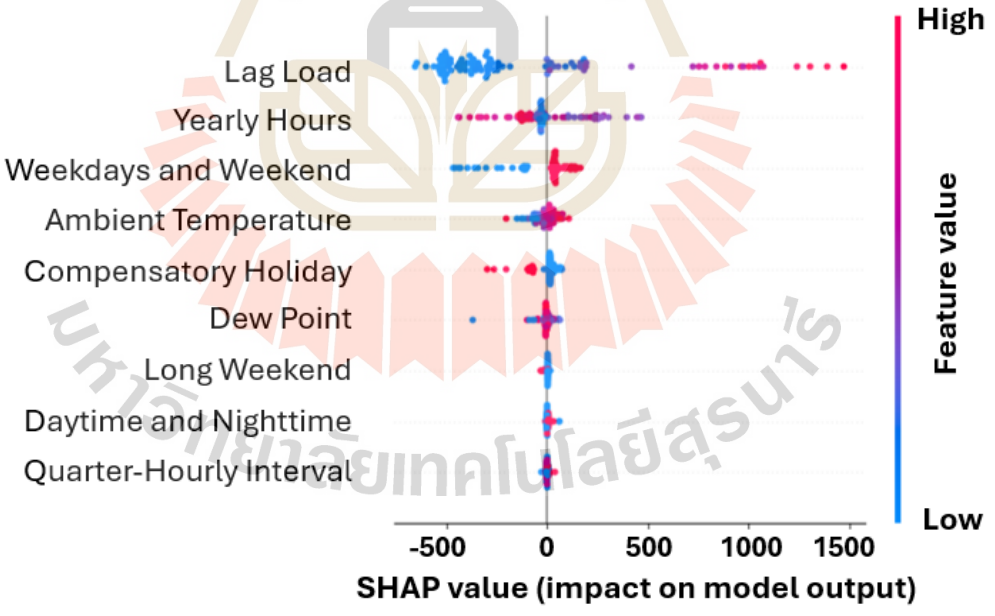


(d)

Figure D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

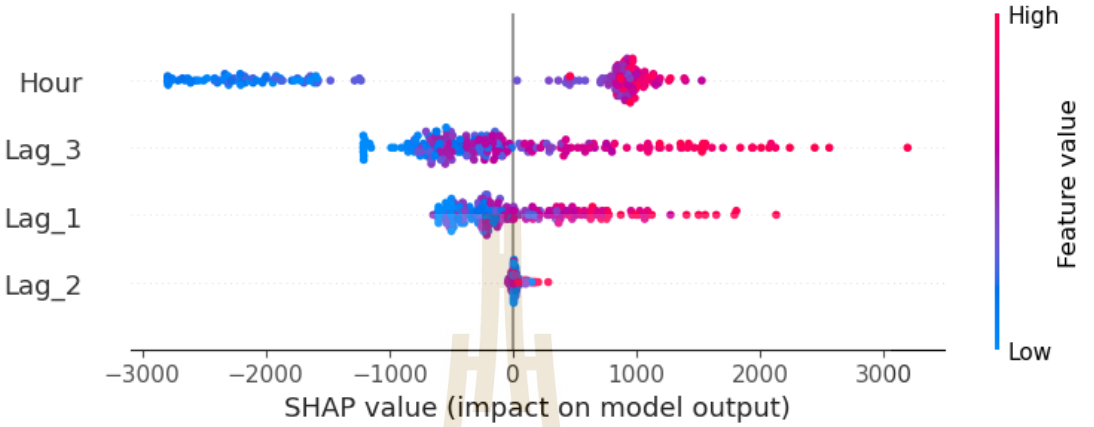


(e)

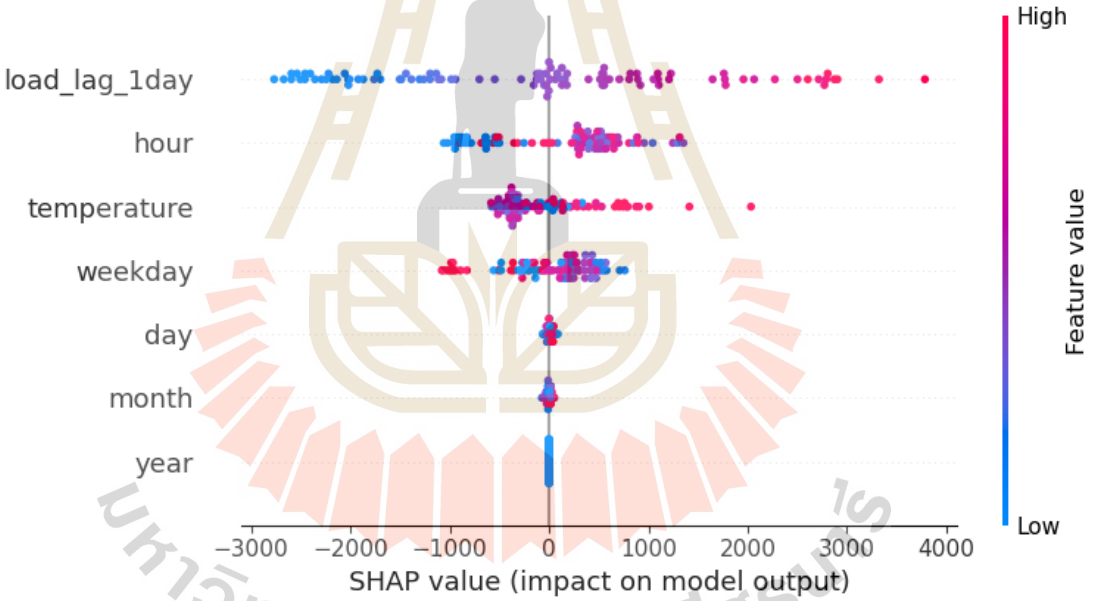


(f)

Figure D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

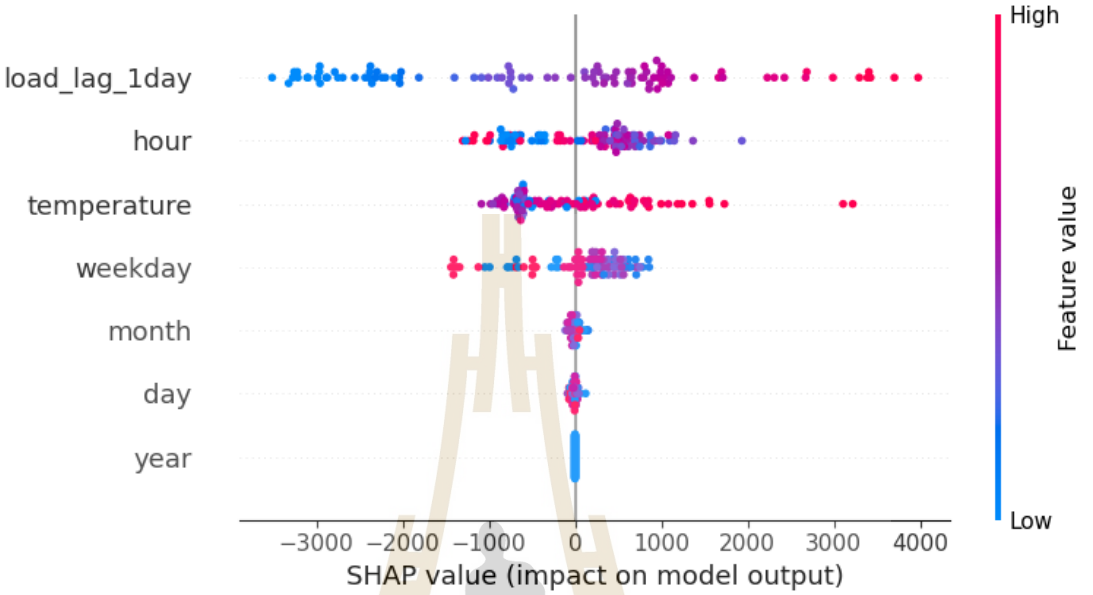


(a)

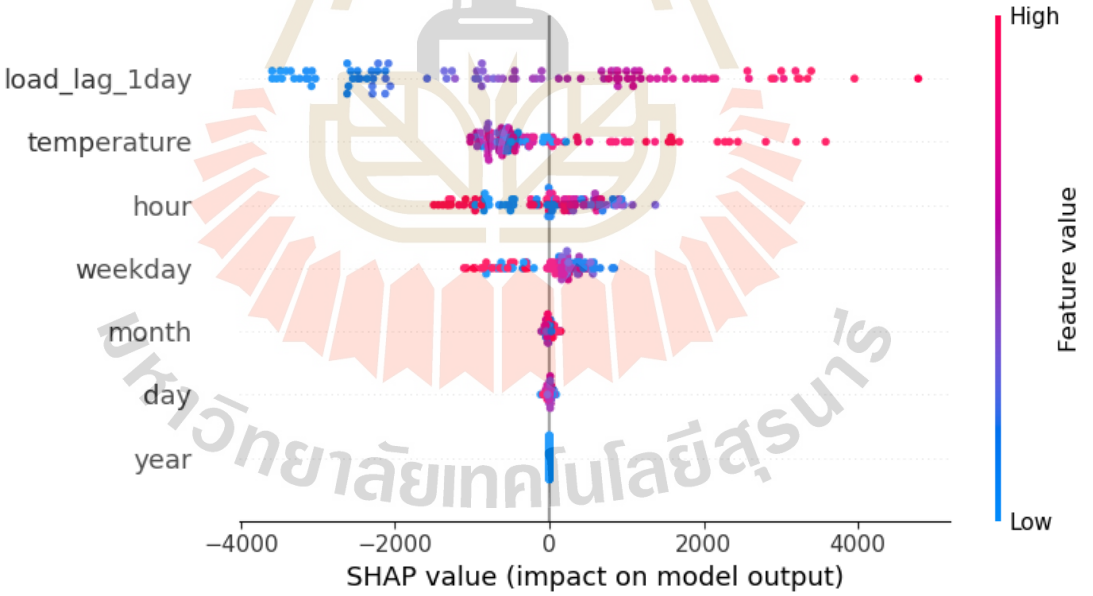


(b)

Figure D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM

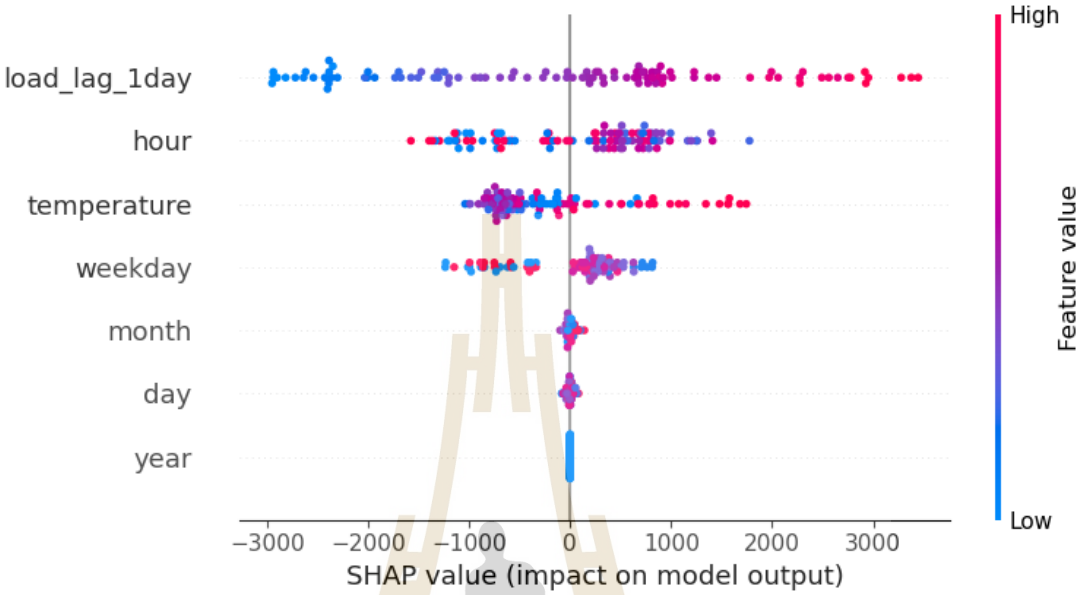


(c)

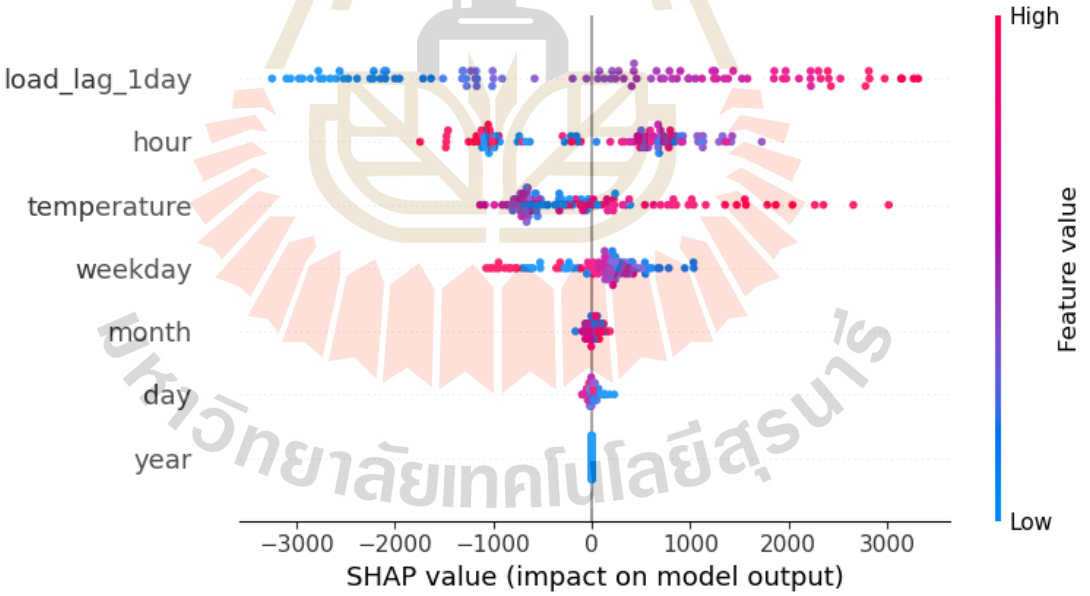


(d)

Figure D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)



(e)



(f)

Figure D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

APPENDIX E

List of publications

N. Thabcha, N. Junhuathon, and K. Chayakulkheeree (2025), Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement: A Case Study of SUT Campus, 2025 13th International Electrical Engineering Congress (iEECON2025), Hua Hin, Thailand.

N. Thabcha, N. Junhuathon, and K. Chayakulkheeree (2025), Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus, ECTI transactions on electrical engineering, electronics, and communications.

มหาวิทยาลัยเทคโนโลยีสุรนารี

Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement: A Case Study of SUT Campus

Nisachon Thabcha
School of electrical engineering
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand
Nisachontabcha@gmail.com

Nitikorn Junhuathon
Department of electrical engineering
Faculty of engineering
Rajamangala University of Technology Thanyaburi
Pathum Thani, Thailand
nitikorn_j@rmutt.ac.th

Keerati Chayakulkheeree*
School of electrical engineering
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand
keerati.ch@sut.ac.th

2025 13th International Electrical Engineering Congress (iEECON) | 979-8-3315-4395-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/iEECON64081.2025.10987599

***Abstract*—This study proposes supervised deep learning (DL) techniques via long short-term memory (LSTM), gate recurrent unit (GRU), Bi-LSTM, and Bi-GRU to forecast electrical loads based on Day-Aware Data Arrangement. The data features are historical electrical loads with arrangement in the form of time series for the model training and evaluation. The proposed model was verified by three accuracy indices: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Square Error (RMSE). Among the study models, the GRU model demonstrates fastest processing speed due to its simpler structure, while LSTM model has the highest forecasting efficiency.**

Keywords—Load forecasting, Long Short-Term Memory, Gated Recurrent Unit, Bidirectional LSTM, Bidirectional GRU

I. Introduction

Efficient energy management, including effective energy planning and load balancing for university or similar facilities, is challenging due to fluctuating or intermittent electrical load data during certain periods. To improve the efficiency of university energy management, accurate short-term load forecasting (STLF) and the development of reliable load forecasting models are essential. However, since traditional forecasting relies on statistical data or overly complex models. Suranaree University of Technology, Nakhon Ratchasima Province, Thailand is an area with continuous and highly fluctuating electricity consumption, both from teaching and various activities held every day. It is necessary to have efficient energy management, reduce unnecessary energy costs through the energy management system (EMS), and help plan and manage energy balance effectively [1]. Short-term load forecasting (STLF) is crucial for EMS, as its accuracy directly impacts system efficiency and performance [2].

Models for load forecasting are crucial, and numerous techniques have been created to achieve this goal. For decades, load forecasting has been a popular topic, and numerous articles using a variety of methods have been published [3], [4]. Regression-based, artificial intelligence, and time-series approaches are the three types of forecasting techniques [5]. ARIMA models and generic statistical analysis are two of the most popular classical time-series techniques. For projecting short-term and medium-term loads, regression techniques are also frequently utilized [6]. These techniques only work with historical load data, are more appropriate for handling load data, and are only applicable to linear data. They might have trouble managing the intricacy of time data, which could result in a large forecasting error for electrical loads [3]. STLF has made extensive use of artificial intelligence and machine learning techniques. These models are capable of handling large and complex datasets and demonstrate adaptability in processing non-linear data. Neural networks (NNs), fuzzy logic systems, and decision trees are a few examples [1]. In electrical load forecasting, NNs are frequently utilized. These are synthetic neural networks designed to resemble the structure of the human brain. The layers that make up their structure are input, concealed, and output layers. They are made to assist in resolving a variety of intricate issues using input data. There are three types of deep learning (DL): supervised, semi-supervised, and unsupervised [7]. A system must learn from the data before it can use supervised deep learning to process a task. The shortcomings of recurrent neural network (RNN) models, particularly with regard to identifying long-term associations in data, are addressed by long short-term memory (LSTM) and gate recurrent unit (GRU) models, which make them better suited for time-sequence data, such as short-term electrical load forecasting with intricate and extremely volatile patterns. Compared to ARIMA, which works well with linear data, they are more adept at handling non-linear and extremely complicated data [8]. Ganguly et al. [2020] presented a comparative analysis of different models for load forecasting using the load profile of a commercial building (amusement park) in San Diego, California, for short-term load forecasting. The results indicate that the LSTM model often yields higher forecast accuracy than the ARIMA and can also be used as part of an EMS. The GRU has a simpler structure than the LSTM model and can be processed quickly, especially for EMS that require real-time response, while the ARIMA may take longer to adjust the model when the data is very volatile [3]. For more accurate forecasting, bidirectional LSTM (Bi-LSTM) and bidirectional GRU (Bi-GRU) models are built from the two models mentioned above. Both of these models perform better than traditional RNN models [9].

Key factors involved in load forecasting, which are considered as load input parameters to improve forecast accuracy and address external influences, include: Temperature, humidity, and weather patterns are important in short-term load forecasting, significantly affecting energy consumption changes. Usage patterns, calendar effects (weekends, public holidays), and social habits are important in capturing energy consumption fluctuations. These input parameters guarantee that the model appropriately adjusts to variations in demand in the actual world [10]. Given the unpredictability and flexibility of renewable energy sources (RESs), it is challenging to estimate consumer demand due to their quick integration into existing power networks. Net forecasting, which uses distinct forecasting procedures for RESs and load, is utilized to get around this problem [7]. By

*Corresponding author

979-8-3315-4395-2/25/$31.00 ©2025 IEEE

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

combining these several forecasts, this method seeks to get the ultimate net forecast. Modern forecasting methods seek to lessen the complexity brought on by external parameter uncertainties by utilizing sophisticated algorithms, real-time load data, and external factors [11].

However, the development of short-term electricity load forecasting models is an important part of the energy management of universities or similar loads, and it is challenging to accurately forecast net electricity demand due to fluctuating or discontinuous electricity load data at certain times. The accuracy of the model is still affected by the characteristics of the data that change over time and days. For example, the energy consumption behavior on weekdays may be different from weekends or public holidays, and it requires a lot of statistical data or complex external factors for traditional forecasting. Therefore, the motivation of this paper is to present a different approach from the traditional method, which is the machine learning or artificial intelligence method using DL through LSTM, GRU, Bi-LSTM, and Bi-GRU models to compare in the same context, making it easy to select the most suitable model and develop the model with the highest accuracy by improving the efficiency of the model through the Day-Aware Data Arrangement approach. It emphasizes that the model can learn the characteristics of the data on each day accurately, reducing the reliance on a large amount of historical load data for forecasting. The accuracy of the electricity forecast results is checked with three accuracy indices: MAE, MAPE, and mean square RMSE, respectively, to solve the above problems.

The highlight of this research is the application of deep learning and Day-Aware Data Arrangement to the university load, which is classified as commercial load, which has fluctuating load usage characteristics because it is a type of load that uses energy related to providing educational services and related activities to create accurate and reliable load forecasting. Short-term electricity forecasting also helps colleges design and improve energy management systems that are more automated and respond to demand more quickly. It also serves as a basis for further research to create smart energy systems in educational institutions and also provides important information for creating appropriate and sustainable energy policies in the future.

This paper is structured into four sections: Section I introduces the study, Section II outlines the methodology, Section III presents the results and discussion, and the final section provides the conclusion.

II. Methodology

A. Data Collection and Preparetion

The datasets collected and used in the forecasting include electric load data from utility meters, which shows historical data from the Automatic Meter Reading (AMR) system maintained by the Provincial Electricity Authority (PEA), which records electricity usage according to load demand at the university, as well as solar energy production data, which records energy production from meters installed in the area with solar panels at Suranaree University of Technology in Thailand. Data regarding compensatory holiday long weekends is gathered from university announcements and the academic calendar. A reliable public website with an API for direct data extraction is the source of historical condition data. Every 15 minutes for a year, from January to December 2023, these statistics are collected and compiled. The electrical load forecasting utilizes 35,040 datasets in total.

In the data cleaning process, which is an important process in preparing various data before using it for forecasting in the model, the following methods are used: Handling missing values (Handling Missing Values) will delete the missing data range or the data range with abnormalities and add new data from the method of calculating the average and transforming the data format to be appropriate (Data Formatting) using Timestamp to help create the Time Factor and Social Factors features, helping the model to learn the relationship of the data better and be more efficient in forecasting.

Since the electrical load in the university is a high PV penetration load with a proportion of solar energy produced and sent to the total power system, the electric load data from utility meters or net load obtained is data that is mixed with solar energy production data or PV generation, making it possible to see electricity usage according to the load demand that is unrealistically low. Therefore, PV Generation is added back to obtain pure load or electrical load data that can be used to assess the actual electricity demand in forecasting, as shown in Equation 1.

$$\text{Actual Load} = \text{Metering Load} + \text{Metering PV Generation} \quad (1)$$

The data is divided by taking the entire cleaned dataset for the year and splitting it into three sets including the training set, which uses 80% of the total data; the validation set, which uses 20% of the data for parameter tuning; and the test set, which consists of a separate portion of data that the model has never seen before, reserved for final evaluation.

B. Feature Extraction

Assigning features to existing data is very important in forecasting electrical load. Because features are variables that will be fed into the model to help the model learn relationships and generate accurate forecast results. Features used in forecasting this type of data are a collection of parameters that are significant to load forecasting and are created by taking into account the context of time and day. Creating features from data can be divided into two types as follows: The first type is creating features that are Day-Aware Data Arrangement, which is a method of arranging data that takes into account the context of time and day to allow the model to learn the characteristics of the data on each day or specific time period. It consists of Hour of Day, which is a value that specifies the hours in one year; Quarter-Hourly Interval, which is a value that specifies the minutes every 15 minutes in each hour; Weekdays and Weekend, which are values used to separate data between weekdays and weekends; Daytime and Nighttime, which are values used to separate data between day and night; Compensatory Holiday, which is a value that specifies whether that day is a holiday or not, Long Weekend, which is a value that specifies that period is a special holiday or festival. The second type is a feature that is weather data, consisting of ambient temperature data and relative humidity data. The number of data sets used to create and divide the feature data sets will be the same as the data used in forecasting electrical load.

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

C. LSTM and Bi-LSTM architecture

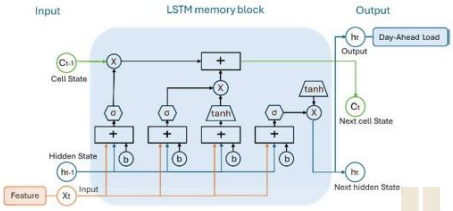


Fig. 1. LSTM architecture.

LSTM is a special kind of RNN that is capable of identifying long-term dependencies. LSTMs are purposefully designed to avoid the problem of long-term reliance. They essentially behave in such a way that their default state is long-term memory retention [12]. For time series forecasting, they are hence the perfect parallel [13]. Fig. 1 describes how LSTM cells are represented graphically.

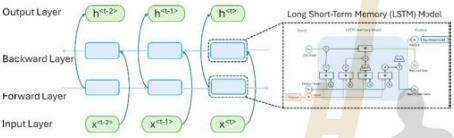


Fig. 2. Bi-LSTM architecture.

Bi-LSTM is a sequence processing model that is made up of two LSTMs, one of which accepts input in a forward direction and the other in a backward manner. Bi-LSTMs give the network access to additional information, which improves the context of the algorithm [14]. The primary advantage is that each input sequence piece includes both current and historical data. Therefore, Bi-LSTM can produce a more pertinent outcome by merging LSTM layers from both directions [15]. The configuration of LSTM layers in Bi-LSTM is shown in Fig. 2.

D. GRU and Bi-GRU architecture

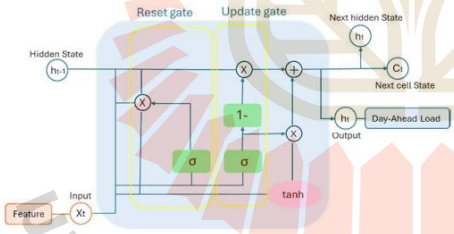


Fig. 3. GRU architecture.

GRU resemble LSTM, which regulates data flow using gates. But with a simpler design, GRU provides an improvement over LSTM. GRU just has a hidden shape, not a distinct cell state like LSTM. The GRU has unrestricted access to all concealed states. Compared to LSTM units, GRUs are easier to train and do not require a memory unit to regulate the data flow [16]. As illustrated in Fig. 3, GRU may train more quickly or employ less data to generalize since it has fewer parameters.

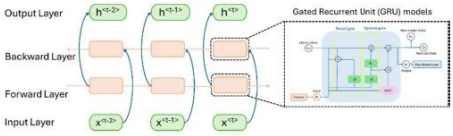


Fig. 4. Bi-GRU architecture.

Bidirectional GRU - A bidirectional GRU, often known as a Bi-GRU, a bidirectional GRU is designed to analyze univariate time series data. This enables the temporal dependencies found in the electrical load data to be accommodated. This integration becomes essential for precise forecasting by utilizing data from both previous and subsequent time periods. As illustrated in Fig. 4, this functionality plays a substantial role in improving forecasting precision [17].

E. Load Forecasting Framework

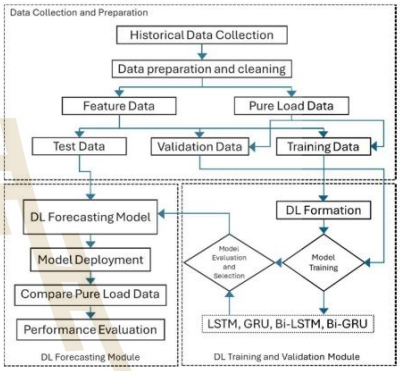


Fig. 5. The Proposed Load Forecasting Framework

The electrical load forecasting, including finding the model with the appropriate accuracy for the university's load data, can be done as follows. Start by collecting all relevant historical data, then clean the data with necessary techniques to facilitate model learning. The data is divided into three groups: the training data group receives feature data as input and pure load data for the model to learn the relationship, then performs the deep learning formation step, which is to design the structure and select the function for the various models and train the model. The models used for training include LSTM, GRU, Bi-LSTM, and Bi-GRU, and they output the load forecast. The validation data group receives the same input as the first group, receiving various models for forecasting. When the output is the load forecast, it will help check the accuracy of model training and prevent overfitting in the model training process. It also helps evaluate the performance of the model during model training of the predicted values and actual load values. Then select the best-performing model as the deep learning forecasting model. The test data group receives features as input, inputs data into the selected model, and outputs daily load forecasts and obtains the developed model. By comparing the predicted values with the actual load values available, the performance of the tested model can be evaluated, as can be seen in Fig. 5.

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

F. Performance Factor

To examine the performance of the proposed forecasting model through evaluation metrics including percentage of MAPE, MAE and RMSE, as shown in Table I.

TABLE I. PERFORMANCE FACTOR

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - y_i')}{y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^{n}\lvert y_i' - y_i\rvert}{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2}$

III. RESULTS AND DISCUSSION

Python 3 software was employed to simulate the Suranaree University of Technology's daily electrical load forecast in order to create a model suitable for the university's electrical loads and evaluate and contrast efficiency using evaluation metrics including MAPE, MAE, and RMSE. The data selected for use was every 15 minutes throughout 2023 and contained, in addition to pure load and feature or electric load data used for forecasting, The models made use of to simulate the results were LSTM, GRU, Bi-LSTM, and Bi-GRU. To get this data ready for entry into the model, it was separated into three sections. This section will show the outcomes of the simulation and different model settings.

TABLE II. MODEL PARAMETER SETTINGS.

Methods	LSTM	GRU	Bi-LSTM	Bi-GRU
Layer	2	2	2	2
Units	100	100	100	100
Dropout rate	0.2	0.2	0.2	0.2
Learning rate	0.001	0.001	0.001	0.001
Batch Size	4	4	4	4
Epochs	50	50	50	50
Optimizer	Adam	Adam	Adam	Adam

TABLE III. EVALUATION RESULTS OF MODEL PERFORMANCE IN VALIDATION METRICS AND MODEL TRAINING TIME.

Methods	Validation Metrics			Model Training
	MAPE (%)	*MAE (kW)*	*RMSE (kW)*	*Time (s)*
LSTM	16.36	227.48	428.84	296.34
GRU	20.61	271.96	464.28	176.76
Bi-LSTM	16.86	241.22	439.58	589.14
Bi-GRU	17.71	252.18	453.94	318.95

A small data set of size four is used to improve the diversity of learning, using 50 epochs to give the model enough time to learn from the training data, and the model structure is set to have 100 units as the number of units in the hidden layer to train different DL models. Two layers of each model are used, a dropout rate of 0.2 is used to prevent overfitting, and Adam Optimizer is used to adjust the model weights, and a learning rate of 0.001 is used as the learning rate for Adam Optimizer, as shown in Table II. The GRU model trained faster than the LSTM model, which took 296.34 seconds. Along with the evaluation of the model's performance in the validation set data, the model training time could indicate the device's usage of resources. The findings of the model's electrical load forecasts will be validated using this data. With MAPE = 16.36%, MAE = 227.49 kW, and RMSE = 428.85 kW, the LSTM model performs best after the performance evaluation. The Bi-LSTM model, with MAPE = 16.86%, MAE = 241.22 kW, and RMSE = 439.5889 kW, is the experimentally efficient model. The two models' training errors are comparable to those of the GRU model (MAPE = 20.61%, MAE = 271.96 kW, and RMSE = 464.29 kW) and Bi-GRU (MAPE = 17.71%, MAE = 252.19 kW, and RMSE = 453.94 kW), according to the three metrics. According to the validation dataset, results demonstrate the extent to which the LSTM and Bi-LSTM models learn and capture relationships. With its speed and predicting accuracy, LSTM offers of potential for real-world use. Table III displays the values in the validation set, which includes the model training settings or the simulation results of the validation metrics' performance.

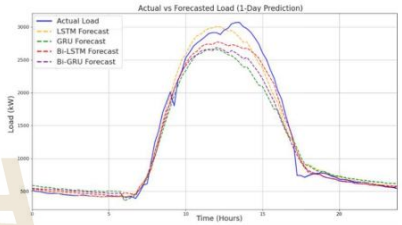


Fig. 6. One-day-ahead electrical load forecast for each model on weekdays

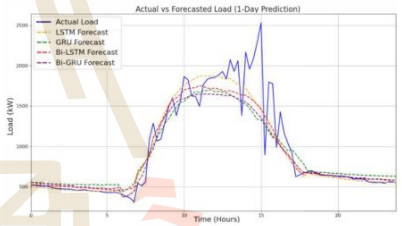


Fig. 7. One-day-ahead electricity load forecast for each model on weekends

After model training and validation, the model is tested in real use. The test set data is divided into two types: weekday load is the load from Monday to Friday, which has a higher total electricity demand than the holiday load. Additionally, weekend load is the load on Saturday or Sunday with the largest variability in electricity demand. The following test days have been selected to evaluate the precision and capacity for forecasting each load model when the electrical load exhibits distinct characteristics: The weekday load is Wednesday, March 1, 2023. And the weekend load is Sunday, October 1, 2023. Forecasting the electrical load output for a day or 24 hours in the future is required, with data being displayed every 15 minutes. There will be a total of 96 data sets. The actual test date is simulated through each model. The results obtained from simulating weekday load are shown in Fig. 6 and the results from simulating holiday load are shown in Fig. 7. The forecast graph will appear. The blue line represents the actual electrical load, or pure load, the yellow line, the green line, the red line, and the purple line. Shows the

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

forecast of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. In the weekdays load forecast, the LSTM model graph gives the forecast that is closest to the actual load graph. The graph is closely related to the actual load graph. The next order has the forecast that deviates from the actual load, which are the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively. However, other models can provide the forecast of the actual load trend. In the part of the weekend's load forecast, the LSTM model graph still gives the forecast that is closest to the actual load graph. However, there are still some load periods that do not give the forecast that is as close to the actual value as it should be because it is a day with a significant load fluctuation. Overall, the LSTM model still performs well. The models whose forecast graphs are closest to the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively.

TABLE IV. PERFORMANCE OF EACH MODEL ON WEEKDAY LOAD

Methods	Model Performance of Weekday Load		
	MAPE (%)	*MAE (kW)*	*RMSE (kW)*
LSTM	6.09	81.51	122.84
GRU	13.29	170.33	242.46
Bi-LSTM	7.38	91.38	131.30
Bi-GRU	9.88	135.16	196.28

TABLE V. PERFORMANCE OF EACH MODEL ON WEEKEND LOAD

Methods	Model Performance of Weekend Load		
	MAPE (%)	*MAE (kW)*	*RMSE (kW)*
LSTM	9.62	115.12	214.53
GRU	15.51	157.18	254.17
Bi-LSTM	11.21	127.00	219.68
Bi-GRU	11.42	138.39	243.42

In addition, the performance of each model was evaluated through all three indicators or through evaluation metrics. The results of the weekday load evaluation are as follows: The LSTM model has a MAPE = 6.09%, MAE = 81.51 kW, and RMSE = 122.84 kW. The GRU model has a MAPE = 13.29%, MAE = 170.33 kW, and RMSE = 242.46 kW. The Bi-LSTM model has a MAPE = 7.38%, MAE = 91.38 kW, and RMSE = 131.30 kW. And the Bi-GRU model has a MAPE = 9.88%, MAE = 135.16 kW, and RMSE = 196.28 kW. Shown in Table IV. The results of the holiday load evaluation are as follows: The LSTM model has a MAPE = 9.62%, MAE = 115.12 kW, and RMSE = 214.53 kW; the GRU model has a MAPE = 15.51%, MAE = 157.18 kW, and RMSE = 254.17 kW; the Bi-LSTM model has a MAPE = 11.21%, MAE = 127.00 kW, and RMSE = 219.68 kW; and the Bi-GRU model has a MAPE = 11.42%, MAE = 138.39 kW, and RMSE = 243.42 kW. Shown in Table V. From the evaluation of the model performance, the results of the weekday load, the LSTM model shows the lowest forecast error when compared to the Bi-LSTM model, Bi-GRU model, and GRU model. For the results of holiday load, the LSTM model still has the lowest error of all metrics when compared to other models. However, it still has a higher error than weekday load because it is a day with high volatility, especially in hours 13 to 18, which has the highest volatility, resulting in the highest forecast error during that period. Overall, the LSTM model still has the best power demand prediction and gives the lowest error for that day. The results show that the most efficient and accurate model for university load data is the LSTM model because the model can learn the patterns and relationships of long-term periods well, including managing the time series data in the data set most efficiently. The weekday load pattern has slightly higher forecast accuracy than the holiday load because the holiday load is more volatile. The next most efficient models are the Bi-LSTM and Bi-GRU models because the type of data that needs to be forecasted may not give enough weight to consider the data from the past and the future, causing the model to learn the relationship and not be able to use its full potential. The last model is the GRU model because it is a simpler model than the LSTM model, which makes it not as good at learning the relationship of the data set, and this type of data may not be suitable for the model. While Suranaree University's power load can be reliably predicted by LSTM and Bi-LSTM models, there are still some restrictions that should be considered when using these models in other situations. To properly understand long-term correlation patterns, the models need a lot of high-quality data. In the case of data-constrained environments, such as small-scale power systems in rural areas or organizations with data shortages, the model performance may be reduced. In addition, Bi-LSTM and Bi-GRU models are complex and require higher computing resources than GRUs, which may not be suitable in systems with limited computing resources, such as IoT devices or edge computing systems. However, the models can be applied in different environments, both in the large-scale energy industry and in distributed energy systems, by adjusting the model structure and parameter settings to suit the data complexity in each context or adding variables such as weather data or sunrise-sunset times to increase the accuracy of predicting fluctuating energy usage. The use of GRU models with higher model training speed in real-time systems may be a suitable option. The results of this study have important implications for the energy sector in many dimensions. The developed electricity load forecasting model can help enhance the efficiency of energy management strategies in various contexts. By precisely predicting energy demand ahead of time, energy suppliers may effectively plan power generation and distribution, minimize overloading, and minimize transmission line losses through integration with smart grid energy management technologies. These models can be used in industrial energy management to predict the energy demand for pieces of equipment or production processes. This helps energy managers plan when to use energy (demand-side management), lower electricity costs during periods of high energy prices, and make the production system more sustainable. In addition, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources such as solar and wind power, which can help reduce energy imbalances and improve the efficiency of renewable energy use, as well as support the reduction of unnecessary energy use as part of the strategy to reduce greenhouse gas emissions in the energy sector.

IV. CONCLUSION

This article presents an alternative approach to developing short-term electric load forecasting models and improving their accuracy through deep learning techniques, including the Day-Aware Data Arrangement method. The most widely used forecasting methods in this field were simulated and compared to evaluate their performance and accuracy. The daily electric load forecasting for Suranaree University of Technology often correlates with past time intervals, allowing LSTM to effectively capture these patterns better than other models. As a result, its load predictions are the closest to the actual values. Adding bidirectional layers to the models did not significantly enhance the results for electric load data. The GRU model,

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

while easier to implement and faster to process, demonstrated lower performance and accuracy compared to the LSTM model for sequential data of this nature. Notwithstanding the encouraging outcomes, this study also discusses significant shortcomings and challenges for further applicability. For maximum performance, high-quality and adequate data are necessary, which might be difficult in contexts with limited resources. Nonetheless, similar models can be used in various energy sectors by reorganizing them, adding new variables, and adjusting the parameter values. The study's findings have important ramifications for the energy industry overall. Planning energy production and distribution effectively, reducing energy losses, and maximizing the integration of renewable energy sources are all made possible by the inclusion of accurate forecasting models, particularly LSTMs, into energy management techniques. Faster GRU models may be suitable for real-time applications where computational resources are limited.

Developing better, scalable forecasting algorithms that adapt to the specific load characteristics of universities and can be applied accurately in other contexts will be a major focus of future research. There is much hope for the design and improvement of energy management systems by combining cutting-edge machine learning techniques with real-time adaptive capabilities. This study provides a basic framework for the development of energy management systems and supporting the growth of smart and sustainable energy infrastructure.

REFERENCES

[1] Ahmad, T., Chen, H., Guo, Y., Wang, J., Shair, J., & Al-Ghamdi, S. G. (2018). Load forecasting for a campus university using ensemble methods based on regression trees. Energies, 11(8), 2038. MDPI. https://doi.org/10.3390/en11082038

[2] T Shepero, M., van der Meer, D., Munkhammar, J., Widegren, K., & Widen, J. (2019, June 23–27). Short-term load forecast for energy management system using neural networks with mutual information method of input selection. 2019 IEEE Milan PowerTech, Milan, Italy, 1–6. IEEE Xplore. https://doi.org/ 10.23919/SICEISCS.2019.8758714

[3] Ganguly, S., Nath, D., & Mandal, K. K. (2020). Bone Fracture Detection and Classification using Deep Learning Approach. 2020 International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control (PARC), Mathura, India, 1–5. IEEE Xplore. https://doi.org/10.1109/PARC49193.2020.236611

[4] Taylor, J. W., & McSharry, P. E. (2007). Short-term load forecasting methods: An evaluation based on European data. IEEE Transactions on Power Systems, 22(4), 2213–2219. DOI: 10.1109/TPWRS.2007.907583

[5] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. PLOS ONE, 13(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

[6] Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. IEEE Transactions on Power Systems, 16(1), 44–55. DOI: 10.1109/59.910780

[7] Vanitha, R., & Jothi Swaroopan, N. M. (2023). Contrast of stacked GRU-RNN and bidirectional LSTM network-based process for forecasting renewable energy for smart grid. 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE). Chennai, India, 1–6. IEEE Xplore. https://doi.org/10.1109/RMKMATE59243.2023.10369285

[8] Doe, J., Smith, A., & Brown, C. (2022). A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing. Energy Systems Research Journal, 18(4), 56–72. https://doi.org/ 10.1109/SAS48726.2020.9220021

[9] Prakash, S., Jalal, A. S., & Pathak, P. (2023, March 3–4). Forecasting COVID-19 pandemic using Prophet, LSTM, hybrid GRU-LSTM, CNN-LSTM, BiLSTM, and stacked-LSTM for India. 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India. https://doi.org/10.1109/ISCON57294.2023.10112065

[10] Ahmad, N., Ghadi, Y., Adnan, M., & Ali, M. (2022). Load forecasting techniques for power system: Research challenges and survey. IEEE Access. Received May 30, 2022, accepted June 28, 2022, published July 1, 2022, current version July 11, 2022. https://doi.org/10.1109/ACCESS.2022.3187839

[11] Patel, R., Sharma, K., Gupta, L., & Singh, P. (2024, January 27–29). A comparative analysis of deep learning-based load forecasting techniques. 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control (PARC), Mathura, India, 1–6. IEEE Xplore. https://doi.org/ 10.1109/AISP61396.2024.10475303

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. https://www.deeplearningbook.org/

[14] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681. https://doi.org/10.1109/78.650093

[15] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5–6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

[16] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv. https://doi.org/10.48550/arXiv.1406.1078

[17] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv. https://doi.org/10.48550/arXiv.1412.3555

Biography



Nisachon Thabcha received her Bachelor of Engineering degree in Electrical Engineering (Second Class Honors) from Suranaree University of Technology, Thailand in 2024. She is currently pursuing her Master's degree in Electrical Engineering at the Faculty of Electrical Engineering, Institute of Engineering, Suranaree University of Technology. Her research interests include deep learing technique application in power system and related fields.



Nitikorn Junhuathon was born in Nakhon Ratchasima in Thailand, on February 23, 1995. He received his B. Eng., M. Eng., and Ph.D. In Electrical Engineering from Suranaree University of Technology (SUT), Thailand, in 2017, 2019 and 2023, respectively. He is currently a lecturer at Department of Electrical Engineering, Faculty of Engineering, Raja Mangala University of Technology Thanyaburi (RMUTT), Thailand. His research interests are machine learning and optimization technique for power forecasting, electric vehicle application, and related fields.



Keerati Chayakulkheeree received his B. Eng. in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL) in 1995, and M. Eng. and D. Eng. degrees in Electric Power System Management from the Asian Institute of Technology (AIT), Thailand, in 1999 and 2004, respectively. He is an Associate Professor at the School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand. His research interests include power system optimization, AI applications in power systems, smart grid technology, power system protection, distributed energy resources, electricity market restructuring, renewable energy integration, and carbon emissions in the power sector.

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus

Nisachon Thabcha[1], Nitikorn Junhuathon[2,*], and Keerati Chayakulkheeree[1,*]

[1] School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology
[2] Department of Electrical Engineering, Faculty of engineering, Rajamangala University of Technology Thanyaburi
Email: nisachontabcha@gmail.com, nitikorn_j@rmutt.ac.th, keerati.ch@sut.ac.th
* Corresponding Author

ABSTRACT

This research proposes an approach to develop a one-day-ahead electrical load forecasting model using deep learning techniques. It compares deep learning models with basic models, namely Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM, to improve accuracy and reliability through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis. This approach utilizes data collection and feature engineering to optimize the context of an electrical loads with integrated solar power generation, Using Suranaree University of Technology (SUT) campus electrical system as a case study. Three accuracy indices are examined: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). Among the NN-based models, Bi-LSTM and LSTM tend to provide the best overall forecasting performance from the input data, and the Bi-LSTM model is the most accurate model with the lowest metric value among all models. The lag load feature, or the value of the electrical load in the past one day, is the feature that is most related to the forecasting target and has the most impact on the model's decision.

Keywords: Load Forecasting, Long Short-Term Memory, Bidirectional Long Short-Term Memory, Mutual Information, SHAP, Explainable AI

1. INTRODUCTION

It is well known that electric power is one of the most widely used energy sources and has been used for a long time, both in people's daily lives and well-being and as an important part of national and economic development [1]. The greater the economic growth, the greater the population's demand for electricity [2]. Electric power must, therefore, be reliable and managed with quality [3]. In Thailand, there has been a continuous increase in renewable energy and the introduction of electric vehicles, resulting in unpredictable patterns of electricity usage, with fluctuations that are difficult to manage [4]. Therefore, developing accurate and transparent electricity load forecasting is important for managing electricity to achieve maximum efficiency. It is also linked to the National Energy Plan (NEP2024), which defines five pillars of electric power as follows: Pillar 1: Demand Response & Energy Management System (DR & EMS). Load forecasting is the basic information that the EMS system uses to plan production and load management in advance. This includes coordinating the load response for maximum efficiency, reducing peak demand, and increasing the stability of the power system. Pillar 2: Renewable Energy Forecast (RE Forecast), load forecasting and renewable energy are in tandem in maintaining the power system balance, especially when renewable energy is highly volatile. Accurate load forecasting will reduce the uncertainty in power dispatch and support the adoption of a higher proportion of renewable energy into the system. Pillar 3: Microgrid & Prosumer Local Energy Systems, load forecasting in this context enables the appropriate management of energy resources in each area, reducing the risk of power outages or overloads. Pillar 4: Energy Storage System (ESS) and Pillar 5: EV Integration, although not directly related, load forecasting helps to manage charging and energy storage or discharge behaviour more efficiently. If energy storage systems (batteries) are installed in the future or electric vehicles (EVs) are used at a high level, load forecasting will play a role in designing systems to support charging and energy usage at different times [5].

There are three popular methods for forecasting electricity load. Regression-based methods are considered simple models, such as linear regression, which work well with statistically recorded data and determine the relationship between inputs and outputs. However, complex and non-linear energy demand data used for forecasting can pose problems. Time series methods, typically used with time-dependent data, such as ARIMA models, face challenges when load changes suddenly and due to external factors, making the original data difficult to predict and uncertain [6]. Finally, artificial intelligence (AI) and deep learning (DL) techniques are used. These models are more adaptable to large, voluminous, volatile, and complex data sets. Models belonging to the Neural Network (NN) family are currently trending towards widespread use in forecasting [7]. Table 1 presents a summary of key issues of each research study, including feature engineering, selecting the best model from a comparison of forecasts with other models, metrics used for evaluation, techniques used for feature selection, and

Table 1: *Review the Literature and Summarize the Objectives*

Ref	Best Deep Learning Model	Feature Selection	Model Explainability	RMSE	MAE	MAPE
[4]	LSTM	MI	-	✓	✓	-
[8]	LSTM, Bi-LSTM	-	-	✓	✓	✓
[9]	-	-	SHAP value based	-	-	✓
[10]	RAID	-	SHAP	-	-	✓
[11]	-	Correlation	KAN	✓	-	-
[12]	LSTM-Attention-LSTM	Attention weights	Attention weights	✓	-	✓
[13]	Attention-LSTM	Attention weights	Attention weights	✓	✓	-
[14]	CNN-LSTM	Correlation	-	✓	✓	✓
[15]	Bi-LSTM	MI	-	✓	-	✓
[16]	ANN	MI	-	-	-	✓
[17]	MLP	-	SHAP	✓	-	✓
Proposed Framework	**6 models**	**MI**	**SHAP**	✓	✓	✓

model explainability, to address gaps in research and the objectives of this study.

However, it has been found that many works employ deep learning models in the context of load forecasting. Deep learning models are more effective at learning temporal relationships and non-linear behavior than traditional load forecasting methods, coupled with the selection and design of contextual feature engineering.

However, there is a lack of comparisons between diverse models, including traditional methods and deep learning or NN models, including feedforward models compared to RNNs, models that incorporate RNN models, and hybrid models. Furthermore, there is a gap in predicting features that are effective for nonlinear data, clearly explaining the results of black-box models, and identifying features that truly influence the model's priorities. The trend in load forecasting model development in research focuses on interpretability to enhance both efficiency and transparency. Many studies have focused on using SHAP and other Explainable Artificial Intelligence (XAI) techniques to analyze feature importance, such that the results obtained after model training are designed to directly assess feature importance. However, this lack of integration with methods that filter data before modeling is found. Popular MI methods select accurate features before modeling to reduce model bias. Furthermore, some studies have found that using attention weights can only describe the timesteps that the model focuses on for model explanation. Therefore, our research aims to develop approaches to address these challenges. This leads to the following main objectives:

1) To develop a one-day electricity load forecasting model using deep learning techniques. We will compare deep learning models with basic models, including Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. Specifically, we will examine the model with the highest accuracy based on real-world load data reconstructed from electricity meters and solar panels.

2) To improve accuracy and reliability through the design of appropriate and high-quality features. Mutual Information (MI) is a promising approach for capturing complex nonlinear energy consumption patterns. and systematic evaluation using performance indicators including RMSE, MAE, and MAPE.

3) To improve the transparency and interpretability of the model, SHAP analysis is used to identify influential features and analyze their significance. All features in the dataset can be described to provide a deeper understanding of the model's performance.

This research focuses on the accuracy and reliability of one-day-ahead electricity load forecasting at Suranaree University of Technology (SUT). The project uses real-world load data from the Provincial Electricity Authority (PEA), collected every 15 minutes for a year. Deep learning techniques using artificial neural networks compare models from various categories. Data is prepared, and feature engineering is developed based on the context of the area to reflect the university's actual energy demand. The importance of features can be explained both before and after input into the model for electricity load forecasting. The highlight of this work lies in the systematic design of the forecasting process to achieve both model accuracy and feature importance analysis through the integration of MI and SHAP methods. Most studies use only one technique for feature evaluation and evaluating features with non-NN models due to their inherent black-box nature. However, this work interprets features from deep learning models and conducts two-step feature analysis. The first step, using MI, analyzes the relationship between real data and load, providing a statistical overview. The second step, using SHAP, analyzes the impact on actual model decisions, demonstrating the model's performance on the data, comparing and utilizing the advantages of both methods for optimal forecasting performance. Furthermore, black-box models can be explained and data analyzed before being imported into the model.

At the institutional level, this model provides a more accurate energy planning approach, enabling demand management and cost optimization, as well as more

flexible power system deployment [18]. It also employs a data-driven organization that can support peak demand management [19]. The impact on the energy sector extends beyond university energy management and can be scaled up to other contexts, including smart grid energy management. This approach improves energy management efficiency across various contexts by accurately forecasting energy demand. Energy suppliers can plan energy production and distribution, reduce overload, and effectively reduce energy losses in transmission lines through integration with energy management technologies in smart grids [20]. In the industrial energy management sector, it helps forecast energy demand for equipment or processes, helping energy managers plan when to use energy (demand-side management), reducing electricity costs during periods of high energy prices, and making production systems more sustainable [21]. In the conjunction with the renewable energy sources sector, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources, which can help reduce energy imbalances and improve the efficiency of renewable energy use [22]. In the Greenhouse Gas Emissions sector, it supports reducing unnecessary energy use as part of a strategy to reduce greenhouse gas emissions in the energy sector [23]. This paper is divided into four sections: Section 1 introduces the study and stating the research objectives; Section 2 presents the research methodology used; Section 3 presents the study results and discussion; and Section 4 concludes with a conclusion.

2. METHODOLOGY

2.1 Data Collection and Preparation

Before entering the model for load forecasting, Suranaree University of Technology's electricity consumption data was collected for 1 year, collected every 15 minutes in 2023, totalling 35,040 data sets, which is considered high-resolution data for load forecasting. The data was collected from Automatic Metering (AMR) maintained by the Provincial Electricity Authority (PEA). In addition, solar energy generation data was collected from an Automatic Metering (AMR) maintained by an external company, ENOS, with the same resolution as the electricity load data. Weather data and academic calendar data for that year were collected and prepared to input all this data into the future feature engineering method. The required data is the actual load data. Therefore, the load data obtained from the meter was combined with the solar energy generation data to obtain the data showing the actual load of the university, as shown in Fig. 1.

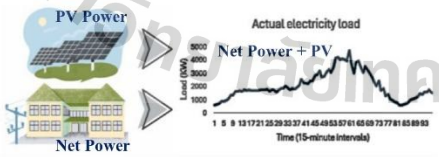


Fig.1: *Preparation of Actual Electricity Load Data*

The preparation of the real electricity load data can be shown on Fig.1. All data used as features and output for the model to learn were divided into two parts: the first part was the data set to be trained, and the other part was partitioned for model validation. Time Series Cross-Validation (TSCV) was performed on this data set, which is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. The training and validation sets were generated using a Time Series Split with 5 folds. The second part was the data set tested on the model after training and validation were completed. It was a data set that the model had never seen before, so the results were unbiased and divided into 4608 data points. For testing, this group of data will be divided into 10 sets; each set will have 3 days. Each set is divided according to the types of data groups, namely long holiday, weekdays during the school term, exam period, summer season, the start of the semester, compensation holiday period, rainy season, weekdays and weekends, term break, and winter season.

2.2 Feature Engineering

In this research, feature engineering has a group of data that has the characteristics of feature creation, which is creating new features from existing time data and converting time data that may not be easy for the model to logical data with contextual meaning. It consists of the following techniques: In the feature transformation method, features that use this method include Weekdays and Weekend, Daytime and Nighttime, Compensatory Holiday, and Long Weekend, which are converting to variable values and binary representation with numbers 0 and 1. In the time-based feature engineering method, features that use this method include yearly hours, which divides the data into meaningful time sequences with 1 year's data every 15 minutes, creating features that allow the model to learn the day and time. The Time Component Extraction method features that use this method include Quarter-Hourly Interval, which extracts time components into separate features, creating new features that describe the sub-patterns that occur in each hour and the lag features method. Features that use this method include Lag Load, which creates features from previous values by taking historical load values to create features. In addition, there is a group of characteristics: external features. External variables affect the forecast of the target variable, such as ambient temperature, an external factor that directly affects energy use behavior, and dew point, an indicator of air humidity.

2.3 Feature Selection and Mutual Information

Mutual Information (MI) captures the relationship of input features to a non-linear data model, shows their importance, and indicates how the feature reduces uncertainty with respect to the forecasting target. It is used as a tool for selecting appropriate forecasting features, especially in tasks where the number of features varies without necessarily being linear. It is calculated using the following Eq.(1).

$$i(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\cdot log\left(\frac{p(x,y)}{p(x)\cdot p(y)}\right), \quad (1)$$

where Y is the target variable, i.e., the future load, X is the input variable, $p(x,y)$ is the joint probability of x and y, and $p(x)$ and $p(y)$ are the marginal probabilities of each variable. The value of MI is expressed as a real number greater than or equal to 0. The higher the value of MI, the more the feature is related to the forecast value. MI can help reduce the size of the data to control both the speed of training and the interpretation of the model. MI is a powerful tool for feature selection for time series models, especially in complex load forecasting tasks that change over time and in different contexts. Selecting appropriate features using MI can make the model more accurate and easier to interpret.

Feature selection in load forecasting each feature plays a physical or behavioural role in load patterns. The Lag Load feature was selected because electrical loads are similar and have daily activities that often repeat within 24 hours, making it a very important representative of daily patterns in forecasting. The Yearly Hours and Quarter-Hourly Interval features help the model understand the time sequence of each day and allow the model to recognize period differences. When used in combination with other features, the model can better learn about daily trends. The Weekdays and Weekend features were selected because electricity usage by electricity consumers differs significantly during holidays and weekdays. The Daytime and Nighttime features were selected because people's behaviour shows higher electricity usage in the morning than at night, so the model should be able to see this difference. The Compensatory Holiday and Long Weekend features have different load patterns during long holidays and special holidays. The Ambient Temperature and Dew Point features are weather-related features that reflect the characteristics of electricity consumers when temperatures and humidity change. They also provide the model with sufficient forecasting capabilities when there is load variation caused by changing weather conditions.

2.4 Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) model is a structure of Recurrent Neural Network (RNN) designed to solve problems and manipulate time-series data. It consists of a Forget Gate that determines which data should be forgotten from the original cell state. Let x_t be the input feature vector at time t, $\sigma(\cdot)$ be the sigmoid activation function, h_{t-1} be the hidden state from time t-1, and b_f be the bias vector of the forget gate, W_f be the learnable weight matrix of the forget gate. The forget-gate output is computed as shown in Eq.(2).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \quad (2)$$

where $f_t \in [0,1]$ is close to 1, indicating retention, while values close to 0 indicate erasure. This helps determine whether data should be retained. The next component, the Input Gate (i_t) and the Candidate Cell State, decides which new data to add to the cell state, as shown in Eq.(3) and Eq.(4).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (3)$$

$$\tilde{C}_t = \sigma(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (4)$$

where i_t controls which new data should be imported, and $\tilde{C}_t$ represents the new data processed (candidate memory). The next component, the Update Cell State, combines the Forget Gate and the Input Gate to create a new cell state, as shown in Eq.(5).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

where the forgotten data is multiplied by f_t, and the new, important data is multiplied by i_t and added together. The final section is the Output Gate (o_t) and Hidden State. These define the data to output as hidden states for future forecasting, as shown in Eq.(6) and Eq.(7).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (6)$$

$$h_t = o_t * tanh(C_t), \quad (7)$$

where h_t is the hidden state, used to predict the electrical load at the next point, as shown in Fig. 2.

2.5 Bidirectional Long Short-Term Memory (Bi-LSTM)

The Bidirectional Long Short-Term Memory (Bi-LSTM) model is a structure that is developed from LSTM to increase the ability to learn sequences of data that have relationships in both the past (past dependencies) and the future (future dependencies) simultaneously. It consists of two LSTM layers that work together: a forward LSTM that processes data from the past to the present and a backward LSTM that processes data from the future back to the present at any time t. The results of both directions are combined into a single hidden state as shown in Eq.(8).

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (8)$$

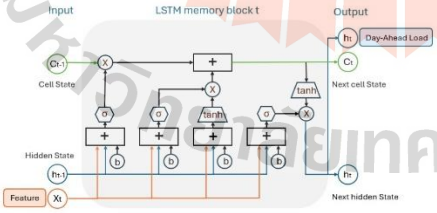


Fig.2: *The LSTM Architecture*

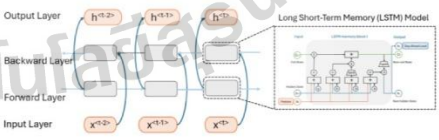


Fig.3: *The Bi-LSTM Architecture*

where $\vec{h}_t$ is the hidden state from the Forward LSTM, $\overleftarrow{h}_t$ is the hidden state from the Backward LSTM, and h_t is the combination of the two results, as shown in Fig. 3.

2.6 Shapley Additive Explanation (SHAP)

SHAP is a method based on the Shapley Value from Cooperative Game Theory, which measures how each feature contributes to the model's decision. SHAP transforms the model description problem into a game in which each feature is a player, and the model's outcome is a reward that must be shared (prediction). The Shapley Value calculation considers all subsets of features (feature coalitions) to assess the impact of each feature. The calculation principle (conceptually) can be described as the Shapley value of feature i is defined by the following Eq.(9).

$$\emptyset i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)], \quad (9)$$

where N is the set of all features, S is the subset of features excluding i, $f(S)$ is the predicted value of the model when using only the feature in S, and $\emptyset i$ is the average of the effects that feature i has on the predicted value when combined with all other features in various orders.

2.7 Load Forecasting Framework

Electrical load forecasting requires the preparation of essential data, including aggregation and collection from reliable sources in the forecasting area, focusing on actual electricity load data from meters. This data is then fed into feature engineering. Two types of data are used: feature generation, which converts raw data into numbers that are easier for the model to understand and allows the model to better learn about data differences, and external feature data, which learns about data trends. This process takes place during the data collection and preparation process. The next step in the simulation process is automatic feature selection using MI to assess the nonlinear relationship between each input variable and the target load. This helps preserve features that are important for forecasting. This reduces data complexity while retaining the most important features. The data is then grouped into three groups for training, validation, and testing, based on proportions. The resulting data is then fed into the model by designing the model, setting various parameters, and training deep learning models in various categories: Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. After training the models on the same dataset for all models, then, in the results and analysis process, the performance is evaluated using standard metrics: root means square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). To enhance the transparency and clarity of the black-box model, the SHAP tool is used to guide decision-making on feature prioritization. This tool identifies the lowest and highest priority features for the final pre-load forecast. Figure 4 shows the entire forecasting process. The first step is the data collection and preparation process. The next step is the simulation process. The last step is the results and analysis process.

2.8 Performance Factor

To examine the performance of the proposed forecasting model through evaluation indicators, namely percentage of MAPE, MAE and RMSE as shown in Table 2, the calculation of the average of each indicator is given by the equation of Average (Avg.), where X is the MAPE, MAE or RMSE indicator on day i and m is the total number of days. Let Y_i denote the actual load at time i, and $\hat{Y}_i$ denote the corresponding predicted load.

Table 2: *Performance Indicators*

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y}_i)}{Y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert}{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$
Average (Avg.)	$\frac{\sum_{i=1}^{m} X_i}{m}$

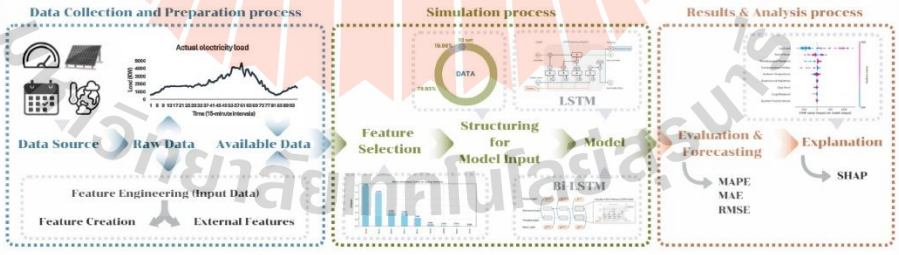


Fig.4: *The Complete Forecasting Process*

3. RESULTS AND DISCUSSION

The electrical load forecast from all the data obtained and the tools mentioned above will be used in simulation using the Python 3 program to show the results comprehensively and explain the accuracy and reliability of the model in the area and context of Suranaree University of Technology. It is a 1-day-ahead load forecast using historical data for 1 year of forecasting in 2023. The details are presented in this section.

The settings for each model are shown in Table 3. All models use similar parameters, namely the Dropout Rate parameter with a value of 0.2 to set the percentage of that unit. To prevent overfitting in the model, the learning rate is set to 0.001 to determine the speed of parameter tuning during learning. The optimizer uses Adam to automatically adjust the learning rate. The batch size, which is the number of data used to calculate the gradient per iteration of parameter tuning, is set to 32. The Epochs, which is the number of iterations to learn the whole data set, is set to 50. The loss function is used to evaluate the model error with MSE. In the different parts, each model is a layer part, which is the basic structure of the model that defines data processing. The DNN model has a Flatten Layer to transform the data and feed it into a Dense Layer, with the number of units in each Dense Layer having the units or the complexity of learning in each layer as 128, 64, and 1. The CNN-LSTM model has Conv1D filters, or the number of filters, as 64. The kernel size is the length of the repeated pattern detection filter with the value of 3. The pool size for reducing the dimensionality is set to 2. The LSTM Units are set to 100 for learning the temporal relationship after Conv1D has extracted features and has a Dense value of 1 in the model part. The Attention-LSTM has an Attention Layer that prioritizes the timesteps that affect the prediction and returns 100 LSTM Units with a Dense of 1. In the LSTM model, the LSTM Unit Capacity is 100, and the Bi-LSTM model has a Bidirectional Wrapper to process two directions and uses 100 LSTM Units in each direction with a Dense of 1.

The selection of the appropriate hyperparameter values for each model was performed using a trial-and-error method. The parameters were adjusted within a specified range and then evaluated by evaluating the minimum MAPE value displayed when each parameter was adjusted within each range. Hyperparameter units were set in a range of 64 to 128, with a best value of 100. The dropout rate was set in a range of 0.1 to 0.3, with the best value of 0.2. The learning rate was set in a range of 0.005 to 0.0005, with a best value of 0.001. Timesteps were set in a range of 4 to 12, with a best value of 8. Batch size was set in a range of 16 to 64, with a best value of 32. Early stopping was set in a range of 5 to 20, with a best value of 10. As shown in Table 4, these hyperparameter sets were then used to simulate the results of each model.

To prioritize the feature data before importing the model with the MI tool, the relationship of the features and the target electrical load dip data groups is measured as both linear and non-linear data. It shows the score ranking of each feature. The Lag Load feature has an MI score of 1.198, the Yearly Hours feature has an MI score of 0.976, the Daytime and Nighttime feature has an MI score of 0.405, the Ambient Temperature feature has an MI score of 0.378, the Dew Point feature has an MI score of 0.241, the Weekdays and Weekend feature has an MI score of 0.055, the Compensatory Holiday feature has an MI score of 0.018, the Long Weekend feature has an MI score of 0.014, and the Quarter-Hourly Interval feature has an MI score of 0.002, as shown in Fig. 5. The introduction of lag load as a feature tends to have a significant relationship between the input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast value. The behaviour of electricity users today and yesterday is similar. The features that are tested with high scores are the Yearly Hours feature, which shows the time position that repeats patterns throughout the year, showing the relationship with the forecast value in a high proportion. These are 2 features that have significantly higher scores than other models. As for other features, the scores are closer to zero and are less relevant to the forecast.

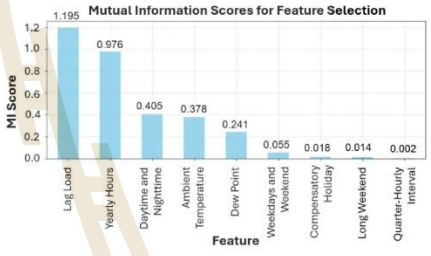


Fig.5: Mutual Information Scores

Table 3: *Parameter Setting*

Model	Parameters Layer
DNN	Flatten, Dense (128), Dense (64), Dense (1), and Units: 128, 64, 1
CNN-LSTM	Conv1D (64), kernel size (3), MaxPooling1D (pool size=2), Dense (1) and LSTM (100)
Attention-LSTM	LSTM (100), return sequences (True), Dense (1) and Attention
LSTM	LSTM (100) and Dense (1)
Bi-LSTM	Bidirectional (LSTM(100)) and Dense(1)

Table 4: Selecting the appropriate hyperparameter

Hyperparameter	Adjustment range	Best parameter values
Units	[64 128]	100
Dropout Rate	[0.1 0.3]	0.2
Learning Rate	[0.0005 0.005]	0.001
Timesteps	[4 12]	8
Batch Size	[16 64]	32
Early Stopping	[5 20]	10

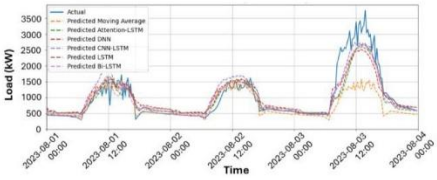


Fig.6: *The Lowest-Performing Forecast Results in the Initial Semester Test Set*

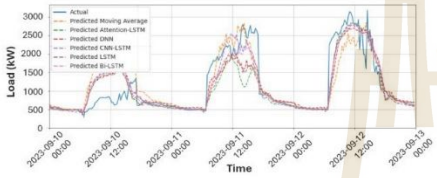


Fig.7: *The Lowest-Performing Forecast Results in the Rainy Season Test Set*

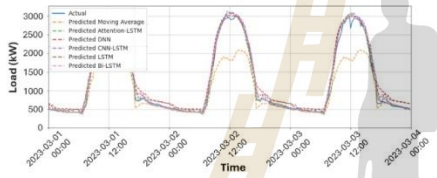


Fig.8: *The Best-Performing Forecasting Results in the Test Set during the Exam Period*

One-day-ahead electrical load forecasting at Suranaree University of Technology can demonstrate the performance of each of the six models through three metrics, compiled into statistical metrics that show the highest, lowest, and average values of each metric from ten test data sets. Important days in the forecast, including the days showing the lowest and highest values of each metric, are also included in the test data set for the model. It is shown in Table 5. The forecasting results show that the moving average forecasting method gives the highest value of all 3 indicators from the tested data set. MAE has a value of 675.72 kW, and RMSE has a value of 945.33 kW on 2023-08-03, which is a date in the set of semesters start dates, as shown in Fig. 6. And MAPE has a value of 78.34% on 2023-09-10, which is a date in the rainy season set, as shown in Fig. 7. The forecasting method with the Bi-LSTM model gives the lowest values of MAE, RMSE, and MAPE indicators from the tested data set, with values of 46.75 kW, 68.35 kW, and 4.82%, respectively, which are a date in the exam set, as shown in Fig. 8. The figures show the load forecast for 1 day in advance for 3 consecutive days. Each model is graphed compared to the actual load graph to show the forecast trend.

Table 5: *Performance of Each Model on 10 Test Sets*

Models	Metric Statistic			
	Metric	Min.	Max.	Avg.
Moving average	MAE (kW)	68.60	675.72	292.63
	RMSE (kW)	87.39	945.33	461.50
	MAPE (%)	5.78	78.34	18.03
Attention-LSTM	MAE (kW)	73.10	385.39	192.96
	RMSE (kW)	114.97	563.92	299.34
	MAPE (%)	5.99	35.40	14.64
DNN	MAE (kW)	97.47	403.80	208.02
	RMSE (kW)	122.82	615.58	308.85
	MAPE (%)	10.16	37.38	17.03
CNN-LSTM	MAE (kW)	68.33	368.25	188.39
	RMSE (kW)	90.55	556.91	293.62
	MAPE (%)	7.21	39.63	14.28
LSTM	MAE (kW)	83.60	362.97	192.74
	RMSE (kW)	105.69	528.20	282.47
	MAPE (%)	9.82	33.84	15.87
Bi-LSTM	MAE (kW)	46.75	373.37	178.63
	RMSE (kW)	68.35	566.79	288.43
	MAPE (%)	4.82	38.74	12.90
Bi-LSTM (Top 3 Features by SHAP)	MAE (kW)	56.95	270.42	183.34
	RMSE (kW)	85.51	388.98	283.21
	MAPE (%)	4.89	36.79	13.99

Since the moving average model relies on the average of historical load patterns, they must be similar at the same time point for accurate forecasting. However, data from the beginning of the semester, the rainy season, or the test data used for forecasting are volatile and unstable. This results in higher forecast error rates than all other models when analyzed for the maximum possible forecast error. The Bi-LSTM model has two layers that process historical data and predict future load trends, enabling it to effectively learn from data patterns that fluctuate with weather and time. Furthermore, the test data during exams or similar data have lag load and yearly hours features that significantly affect the relationship with the forecast at that time, accounting for a significant portion of the MI score. This factor occurs both before and after the forecast point. Therefore, the bidirectional learning method demonstrates the highest forecasting performance compared to all other models when analyzed for the maximum possible model accuracy. The Bi-LSTM model exhibits a lower average error (MAE) value, a percentage error compared to the actual value (MAPE), and a lower proportion of large errors (RMSE) than other models. The model can reduce errors at each time point, reduce forecasts with small errors, and forecast peak load periods more accurately than other models. Therefore, the resulting model is the best-performing model in this context.

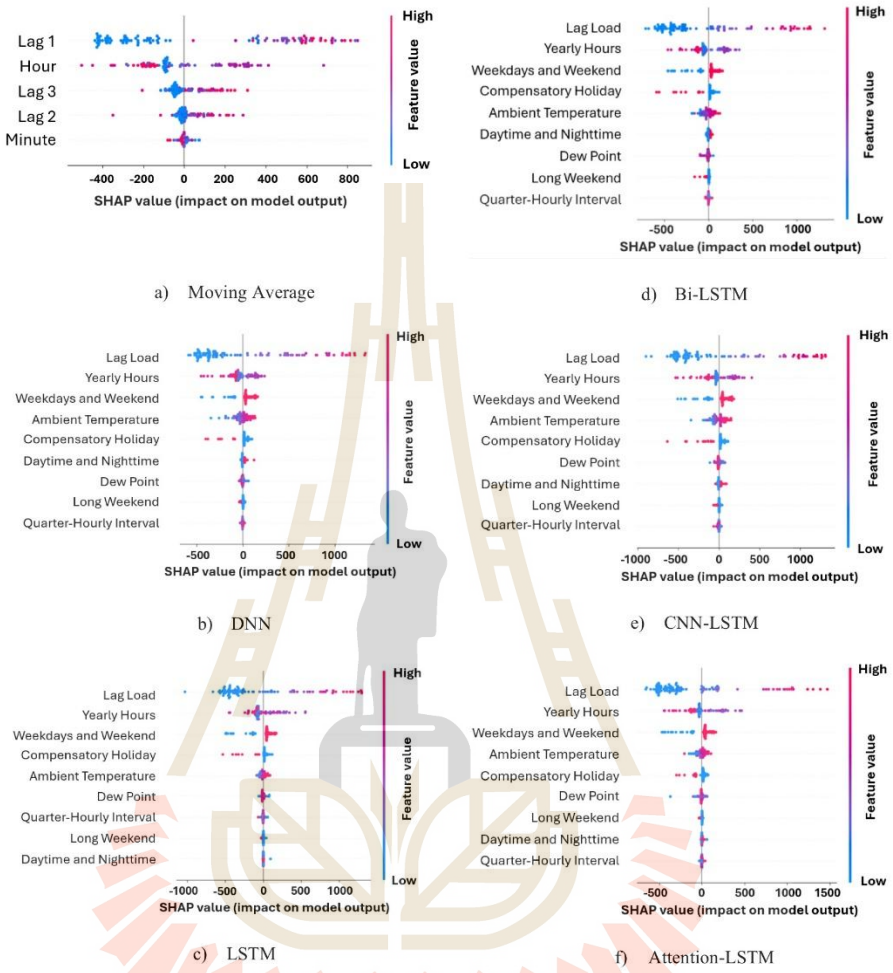


Fig.9: *Shap Value of Each Model:*

a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM.

The LSTM model showed the lowest values of all three indicators as follows: MAE was 83.60 kW, RMSE was 105.69 kW, and MAPE was 9.82%. The model has a learning characteristic from the past to the present. There is a layer that remembers various values, which is different from the DNN model in that the features fed into the model are not chronologically ordered, which makes the indicator values higher than the LSTM model. MAE was 97.47 kW, RMSE was 122.82 kW, and MAPE was 10.16%. For the hybrid CNN-LSTM model, CNN was used to capture short-term data patterns, resulting in lower errors. The lowest values of all three indicators were as follows: MAE was 68.33 kW, RMSE was 90.55 kW, and MAPE was 7.21%. For the Attention-LSTM model, developed from a simple LSTM structure, it prioritizes the most effective timestep, showing an MAE of 73.10 kW, an RMSE of 114.97 kW, and an MAPE of 5.99%. Both models exhibit higher forecasting accuracy than the simple LSTM model. The moving average method yields the best indicators, with an MAE of 68.60 kW, an RMSE of 87.39 kW, and an MAPE of 5.78%. These are more effective than some of the models mentioned above. However, when examining

the worst-day recordings, the model achieves higher indicators than all other models. Furthermore, the average test data also has the lowest overall performance.

The average of the indicators from all 10 test data sets, each set having 3 days or 30 data points, can be shown in order of the models with the lowest to highest average MAE indicators as follows: Bi-LSTM, CNN-LSTM, LSTM, Attention-LSTM, DNN, and moving average models with values of 178.63 kW, 188.39 kW, 192.74 kW, 192.96 kW, 208.02 kW, and 292.63 kW, respectively. The order of the models with the lowest to highest average RMSE indicators is as follows: LSTM, Bi-LSTM, CNN-LSTM, Attention-LSTM, DNN, and moving average models with values of 282.47 kW, 288.43 kW, 293.62 kW, 299.34 kW, 308.85 kW, and 461.50 kW, respectively. The model rankings with the lowest to highest average MAPE metrics are shown as Bi-LSTM, CNN-LSTM, Attention-LSTM, LSTM, DNN, and moving average, with values of 12.90%, 14.28%, 14.64%, 15.87%, 17.03%, and 18.03%, respectively, showing the prediction trend of each model from all the input test data. The Bi-LSTM and LSTM models still have good prediction performance overall.

In addition, the Bi-LSTM model was investigated to obtain the most three influential features from SHAP: Lag Load, Yearly Hours, Weekdays, and Weekend. The lowest values of the metrics were obtained as follows: MAE was 56.95 kW, RMSE was 85.51 kW, and MAPE was 4.89%. Compared to the same model set using all 9 features, the results show that the model using only the most important features provides acceptable forecast performance, although its performance is slightly reduced compared to using all features. The Lag Load feature has the highest weight in describing the daily electrical load characteristics. The Yearly Hours, Weekdays, and Weekend features still help the model better understand hourly and weekly patterns. While selecting only 3 important features allows the model to perform well, using all features yields the lowest metric values and the best performance.

The SHAP tool was imported to describe the importance of features that affect the decision of the moving average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models, respectively, as shown in Fig. 9. The x-axis shows the SHAP value, with a positive value indicating a higher prediction value. The y-axis sorts the features by cluster importance from highest to lowest, and the color of the dots indicates the actual feature value, with red having a high value and blue having a low value. The top three features of the NN model with the highest importance are as follows: The Lag Load feature is an important variable that has a great impact on the model's decision because it has the highest SHAP value distribution. There are a lot of red dots to the right, meaning that if yesterday's load is higher, it will also push today's forecast value higher. The Yearly Hours feature is distributed in clusters and has most red dots to the left, meaning that features with high values will lower the forecast value. The Weekdays and Weekend features show that the closer the SHAP value distribution is to zero, the less the feature's impact on the model's decision. In the non-linear features, the SHAP values are widely distributed, with both positive and negative values and a mixture of red and blue dots at the same point. The moving average method is a basic statistical method for forecasting using features Lag 1, Lag 2, and Lag 3 as the electrical load values for the past 1, 2, and 3 days, respectively. The time-related data, namely the Hour and Minute features, are the time data for all 3 days in every hour and minute at the same time of each day. Feature Lag 1 is the feature with the most dispersion and has the most impact on the model's decision.

The importance of ranking of features of NN models by SHAP and MI, where model A is LSTM, B is Bi-LSTM, C is CNN-LSTM, D is Attention-LSTM, E is DNN, and F is MI. From Table 6, number 1 shows the ranking of the most important features, and the higher the ranking number, the importance will decrease until number 9, which has the lowest feature importance to the model. The Bi-LSTM model forecasts have the highest accuracy. The ranking of features from the highest to the lowest is shown as follows: Lag Load, Yearly Hours, Weekdays and Weekend, Compensatory Holiday, Ambient Temperature, Daytime and Nighttime, Dew Point, Long Weekend, and Quarter-Hourly Interval, respectively, were compared for all 6 models. The Bi-LSTM and LSTM models are models that learn the time sequence of data. The 4th feature is Compensatory Holiday, which shows the compensatory holidays of the university reflect the overall electricity consumption behavior. There are red and blue dots clearly separated from the blue ones. The model can recognize clear patterns with this data and associate it with the load drop. This differs from the fifth-ranked feature, ambient temperature. Since alternating points in the same location have a non-linear relationship, fluctuations can make it more difficult for models to capture these patterns.

Table 6: *Feature Importance Ranking of NN Model by SHAP and MI*

Feature	Feature importance ranking					
	A	B	C	D	E	F
Lag Load	1	1	1	1	1	1
Yearly Hours	2	2	2	2	2	2
Weekdays and Weekend	3	3	3	3	3	6
Compensatory Holiday	4	4	5	5	5	7
Ambient Temperature	5	5	4	4	4	4
Daytime and Nighttime	9	6	7	8	6	3
Dew Point	6	7	6	6	7	5
Long Weekend	8	8	8	7	8	8
Quarter-Hourly Interval	7	9	9	9	9	9

A CNN-LSTM model, which captures short-term and long-term patterns well, or an Attention-LSTM model that emphasizes time steps, will better understand this feature. Based on feature priorities, CNN-LSTM and Attention-LSTM models prioritize ambient temperature over compensatory holidays. DNN models can easily learn by finding association weights, do not remember event sequences, and process data without considering time sequences. The SHAP value of the Ambient Temperature

feature, used to describe load changes, is higher than that of the Compensatory Holiday feature, which relies on time sequences and previous behaviors. The Quarter-Hourly Interval feature, also a later feature, is considered a fundamental feature, as it does not directly correlate with load and is a simple feature for the model to learn. From the simulation results, it was found that the MI method gives more importance to the Daytime and Nighttime, Ambient Temperature, and Dew Point features of each model than the SHAP method because these features have more raw data statistics related to the forecasting target, which reflects the actual behavior of electrical loads without learning the model mechanism and not understanding the relationship between the features like SHAP. When using SHAP, the importance is reduced. For Weekdays and Weekend and Compensatory Holiday features, which have lower MI scores, SHAP gives higher importance to both features. This is because the model learns the effect of these features through the more important Lag Load feature, not just learning the features alone. Using SHAP, one can learn features with complex relationships better.

The third feature of the MI method, Daytime and Nighttime, is shown to be highly correlated with load. However, compared to the SHAP method, this feature is ranked low due to the model's inability to extract patterns well. However, the key features, Lag Load and Yearly Hours, ranked first and second in both methods, showed consistent results, indicating that this feature set can definitely improve accuracy and efficiency. Using load data, the MI method screens out potentially important features related to electrical loads, screening relevant and related data before importing into the model and isolating truly irrelevant features. The trend in importance of features is evident, eliminating bias from the model for optimal forecasting performance. However, since it is not yet possible to determine how features affect model decisions, SHAP values are used at each data point in real-world situations to explain the relationship between features and model decisions. This ensures that the model is indeed using those features and provides a deeper explanation of model behavior, demonstrating the importance of features that directly influence the model. Furthermore, under different load conditions, the direction of their impact on forecasting can be identified. Regarding the reliability of black-box models, this can be explained.

4. CONCLUSION

This research proposes the development of a deep learning-based forecasting technique. The accuracy of the model is compared with that of a baseline NN model. The Bi-LSTM model is shown to have the highest accuracy and performance for load forecasting. The data collected through cleaning and feature engineering are then transformed into features ready for input into the forecasting model. The electricity load forecasts are most closely aligned with actual values. Overall, the Bi-LSTM and LSTM models also yield the lowest metric values, averaging across all tested data, indicating the model's potential for forecasting in this context.

Furthermore, the model reliability investigation indicates that transparent feature analysis and model description using MI can explain the relationships between raw data and electricity loads and filter static data for inspection before inputting the model. This reduces the number of features required for model input and reduces training time. SHAP provides in-depth model descriptions, demonstrating the actual use of the model's features, understanding complex features, and increasing model reliability because all features have underlying relationships that can be demonstrated. Using this combined approach, comprehensive coverage is achieved. It provides accurate forecasting results and can also provide causal explanations.

5. ACKNOWLEDGEMENT

This work is fully supported by Suranaree University of Technology.

6. REFERENCES

[1] International Energy Agency (IEA), World Energy Outlook 2023. Paris, France: IEA, 2023. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2023

[2] U.S. Energy Information Administration, *Electricity explained: Factors affecting electricity consumption*, 2023. [Online]. Available: https://www.eia.gov/energyexplained/electricity/

[3] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2019.

[4] T. Panapongpakorn and D. Banjerdpongchai, "Short-term load forecast for energy management system using neural networks with mutual information method of input selection," in *Proc. SICE International Symposium on Control Systems (SICE ISCS)*, Tokyo, Japan, 2019, pp. 13–18.

[5] Ministry of Energy Thailand, *Thailand Smart Grid Development Plan (2022–2031)*, Bangkok, Thailand, 2022. [Online]. Available: https://www.energy.go.th

[6] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, 2001.

[7] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, Mar. 2018.

[8] S. Atef and A. B. Eltawil, "Assessment of stacked unidirectional and bidirectional long short-term memory networks for electricity load forecasting," *Electrical Power Systems Research*, vol. 189, Dec. 2020.

[9] Y.-G. Lee, J.-Y. Oh, D. Kim, and G. Kim, "SHAP value-based feature importance analysis for short-term

load forecasting," *Applied Science and Convergence Technology*, vol. 18, pp. 579–588, 2023.

[10] J. Jang, W. Jeong, S. Kim, B. Lee, M. Lee, and J. Moon, "RAID: Robust and interpretable daily peak load forecasting via multiple deep neural networks and Shapley values," *Sustainability*, vol. 15, no. 8, 2023.

[11] X. Liu, Z. Yang, Y. Guo, Z. Li, and X. Xu, "A novel correlation features self-assigned Kolmogorov-Arnold Networks for multi-energy load forecasting in integrated energy systems," *Energy Conversion and Management*, vol. 300, 2024.

[12] X. Wen and W. Li, "Time Series Prediction Based on LSTM-Attention-LSTM Model," *IEEE Access*, vol. 11, pp. 48322–48331, 2023.

[13] P. Wu, Z. Huang, Y. Pian, L. Xu, J. Li, and K. Chen, "A Combined Deep Learning Method with Attention-Based LSTM Model for Short-Term Traffic Speed Forecasting," *Journal of Advanced Transportation*, vol. 2020, pp. 1–12, 2020.

[14] K. Ullah et al., "Short-Term Load Forecasting: A Comprehensive Review and Simulation Study With CNN-LSTM Hybrids Approach," *IEEE Access*, vol. 12, pp. 111858–111881, 2024.

[15] S. Hu et al., "Short-Term Load Forecasting Based on Mutual Information and BI-LSTM Considering Fluctuation in Importance Values of Features," *IEEE Access*, vol. 12, pp. 23653–23665, 2024.

[16] R. Azimi, A. Amjady, and A. Daraeepour, "Mutual Information-Based Inputs Selection for Electric Load Time Series Forecasting," *Entropy*, vol. 15, no. 3, pp. 926–942, Mar. 2013.

[17] D. A. Bolstad, U. Cali, M. Kuzlu, and U. Halden, "Day-ahead Load Forecasting using Explainable Artificial Intelligence," in *Proc. IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, New Orleans, LA, USA, 2022, pp. 1–5.

[18] M. Ramya and T. Devaraju, "Dynamic scheduling and islanding energy management in microgrids using deep learning and adaptive reinforcement learning techniques," in *Proc. IEEE International Conference on Intelligent Systems and Sustainable Innovation (ICISS)*, 2025.

[19] P. Shanmugapriya, D. R. P. R., S. V. S., and M. SureshKumar, "Effective approach for energy management system using machine learning algorithm," in *Proc. IEEE International Conference on Power, Energy, Control and Transportation Systems (ICPECTS)*, 2024.

[20] J. Zhang, S. Shi, and N. Zhang, "Mitigation strategies for greenhouse gas emissions and energy consumption based on deep learning," in *Proc. IEEE International Conference on New Perspectives in Smart Industrial Future (NPSIF)*, 2024.

[21] J. D. Joy, M. R. Hossain and A. Chowdhury, "A Hybrid Deep Learning Model for Day Ahead Short-Term Load Forecasting," 2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON), Rajshahi, Bangladesh, 2024, pp. 1-6.

[22] M. A. Hassan, A. A. Eladl, B. E. Sedhom, and M. A. Saeed, "A new optimal centralized demand side management for a campus smart microgrid," in *Proc. IEEE International Symposium on Industrial Electronics (ISIE)*, 2023.

[23] S. Akhtar, M. Z. bin Sujod, and S. S. H. Rizvi, "A novel deep learning architecture for data-driven energy efficiency management (D2EEM) – systematic survey," in *Proc. IEEE International Conference on Engineering and Emerging Technologies (ICEET)*, 2021.

Biography



Nisachon Thabcha received her Bachelor of Engineering degree in Electrical Engineering (Second Class Honors) from Suranaree University of Technology, Thailand in 2024. She is currently pursuing her Master's degree in Electrical Engineering at the Faculty of Electrical Engineering, Institute of Engineering, Suranaree University of Technology. Her current research interests focus on power systems and the application of artificial intelligence in power systems.



Nitikorn Junhuathon was born in Nakhon Ratchasima in Thailand, on February 23, 1995. He received his B. Eng., M. Eng., and Ph.D. In Electrical Engineering from Suranaree University of Technology (SUT), Thailand, in 2017, 2019 and 2023, respectively. He is currently a lecturer at Department of Electrical Engineering, Faculty of Engineering, Raja Mangala University of Technology Thanyaburi (RMUTT), Thailand. His research interests are machine learning and optimization technique for power forecasting, electric vehicle application, and related fields.



Keerati Chayakulkheeree received his B. Eng. in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL) in 1995, and M. Eng. and D. Eng. degrees in Electric Power System Management from the Asian Institute of Technology (AIT), Thailand, in 1999 and 2004, respectively. He is an Associate Professor at the School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand. His research interests include power system optimization, AI applications in power systems, smart grid technology, power system protection, distributed energy resources, electricity market restructuring, renewable energy integration, and carbon emissions in the power sector.

BIOGRAPHY

Miss Nisachon Thabcha received her Bachelor of Engineering degree in Electrical Engineering (Second Class Honors) from Suranaree University of Technology, Thailand in 2024. She is currently pursuing her master's degree in electrical engineering at the Faculty of Electrical Engineering, Institute of Engineering, Suranaree University of Technology. During her graduate studies, she conducted research on short-term electricity load forecasting using supervised deep learning techniques, with a particular focus on time-series modeling, feature engineering, and model interpretability. Her research involved the application of advanced recurrent neural network architectures, including CNN-LSTM and Bi-LSTM, to high-resolution load data combined with temporal and weather-related features.

In addition to her research activities, she served as a teaching assistant for courses related to electrical engineering and power system fundamentals, which enhanced her skills in academic coordination, technical communication, and instructional support. She has presented her research findings at international academic conferences, including the International Electrical Engineering Congress (iEECON), gaining valuable experience in professional scholarly communication.

Her research interests include electricity load forecasting, smart grid applications, data-driven energy management, and interpretable artificial intelligence for power systems. She is passionate about contributing to the advancement of smart grid technology and sustainable energy infrastructure in Thailand and aspires to continue working in this field to support national power system planning and distribution development goals.

DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

Somchat Jiriwibhakorn
.....

(Assoc. Prof. Dr. Somchat Jiriwibhakorn)

Chairperson

.....

(Assoc. Prof. Dr. Keerati Chayakulkheeree)

Member (Thesis Advisor)

.....

(Asst. Prof. Dr. Nitikorn Junhuathon)

Member (Thesis Co-Advisor)

.....

(Prof. Dr. Thanatchai Kulworawanichpong)

Member

Tosaphol Ratniyomchai
.....

(Asst. Prof. Dr. Tosaphol Ratniyomchai)

Member

Yupaporn
.....

(Assoc. Prof. Dr. Yupaporn Ruksakulpiwat)

Acting Vice Rector for Academic Affairs and Quality Assurance

.....

(Assoc. Prof. Dr. Pornsiri Jongkol)

Dean of Institute of Engineering

มหาวิทยาลัยเทคโนโลยีสุรนารี

นิศาชล ทับชา : การพยากรณ์ภาระโหลดล่วงหน้ารายวันโดยใช้เทคนิคการเรียนรู้เชิงลึกแบบมีผู้สอนร่วมกับวิศวกรรมคุณลักษณะเชิงบริบทและการวิเคราะห์แบบอธิบายได้: กรณีศึกษามหาวิทยาลัยเทคโนโลยีสุรนารีที่มีการบูรณาการพลังงานแสงอาทิตย์ (DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร. กีรติ ชยะกุลคีรี, 131 หน้า.

คำสำคัญ : การพยากรณ์โหลดไฟฟ้า, สารสนเทศร่วม, ปัญญาประดิษฐ์ที่อธิบายได้, โครงข่ายประสาทเทียม, อธิบายโมเดลด้วยค่าแชปลีย์

งานวิจัยนี้นำเสนอกรอบการทำงานสำหรับการพยากรณ์โหลดไฟฟ้าระยะสั้น และประเมินประสิทธิภาพของแบบจำลองเชิงลึกโดยใช้ข้อมูลโหลดไฟฟ้าที่มีความละเอียดเชิงเวลา 15 นาที แบบจำลองที่พัฒนาขึ้นใช้ฟีเจอร์ด้านเวลา โหลดไฟฟ้าย้อนหลัง และตัวแปรด้านสภาพอากาศ ร่วมกับกระบวนการสร้างและคัดเลือกฟีเจอร์ที่เหมาะสม แบบจำลองที่นำมาศึกษาเปรียบเทียบประกอบด้วย โครงข่ายประสาทเทียมเชิงลึก โครงข่ายประสาทเทียมหน่วยความจำระยะยาวแบบทางเดียวและสองทิศทาง โครงข่ายประสาทเทียมแบบคอนโวลูชันร่วมกับหน่วยความจำระยะยาว วิธีค่าเฉลี่ยเคลื่อนที่ และโครงข่ายประสาทเทียมที่ใช้กลไกความสนใจ โดยมีการพิจารณาแบบจำลองหน่วยความจำที่มีการควบคุมด้วยประตูแบบทางเดียวและสองทิศทางในการทดสอบเบื้องต้น

เพื่อเพิ่มความแม่นยำและความสามารถในการอธิบายแบบจำลอง งานวิจัยนี้ได้นำวิธีสารสนเทศร่วมและวิธีการอธิบายโมเดลด้วยค่าแชปลีย์มาใช้ในการคัดเลือกฟีเจอร์ การประเมินผลดำเนินการด้วยตัวชี้วัด ผลการทดลองแสดงให้เห็นว่าแบบจำลองที่สามารถเรียนรู้ความสัมพันธ์เชิงลำดับเวลาได้อย่างชัดเจน โดยเฉพาะแบบจำลองหน่วยความจำระยะยาวแบบสองทิศทางและแบบจำลองคอนโวลูชันร่วมกับหน่วยความจำระยะยาว ให้ผลการพยากรณ์ที่ดีกว่าแบบจำลองพื้นฐาน อีกทั้งการคัดเลือกฟีเจอร์ด้วยวิธีการอธิบายโมเดลด้วยค่าแชปลีย์ให้ผลลัพธ์ที่สอดคล้องกับกลไกการเรียนรู้ของแบบจำลองมากกว่าวิธีสารสนเทศร่วม

สาขาวิชาวิศวกรรมไฟฟ้า ลายมือชื่อนักศึกษา นิศาชล

ปีการศึกษา 2568 ลายมือชื่ออาจารย์ที่ปรึกษา

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

NISACHON THABCHA : DAY-AHEAD LOAD FORECASTING USING SUPERVISED DEEP LEARNING TECHNIQUE WITH CONTEXTUAL FEATURE ENGINEERING AND EXPLAINABLE ANALYSIS: A CASE STUDY OF THE SUT CAMPUS WITH PV INTEGRATION

THESIS ADVISOR : ASSOC. PROF. KEERATI CHAYAKULKHEEREE, Ph.D., 131 PP.

Keyword : Load Forecasting / Mutual Information / Explainable AI / Neural Network / SHAP

This study proposes a framework for short-term electric load forecasting and evaluates the performance of deep learning models using electric load data with a 15-minute resolution. The forecasting model incorporates time-related features, historical load data, and meteorological variables, along with feature engineering for effective feature construction and selection.

The evaluated models include Deep Neural Network (DNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Convolutional Neural Network–LSTM (CNN-LSTM), Moving Average, and Attention-based LSTM, with Gated Recurrent Unit (GRU) and Bidirectional GRU (Bi-GRU) considered in preliminary experiments.

To improve forecasting accuracy and model interpretability, Mutual Information (MI) and Shapley Additive Explanations (SHAP) are applied for feature selection. Model performance is assessed using MAE, RMSE, and MAPE. The results demonstrate that models explicitly capturing temporal dependencies, particularly Bi-LSTM and CNN-LSTM, outperform baseline models. Furthermore, SHAP-based feature selection provides feature subsets that better align with the learning mechanisms of deep learning models compared to MI.

มหาวิทยาลัยเทคโนโลยีสุรนารี

School of Electrical Engineering

Academic Year 2025

Student's Signature .....

Advisor's Signature .....

Co-Advisor's Signature .....

ACKNOWLEDGEMENT

The authors are particularly grateful to the Kittibandit Scholarship at Suranaree University of Technology, for which I am sincerely grateful.

I appreciate my advisor, Assoc. Prof. Dr. Keerati Chayakulkheeree, whose guidance, insightful feedback, and constant encouragement have been instrumental in the completion of this work. Beyond academic advice, I am deeply grateful for the valuable life skills and practical wisdom I have gained from him in my daily life. His mentorship has not only shaped my research abilities but also inspired me to develop discipline, perseverance, and a positive attitude in facing challenges both in my studies and in everyday living. I also extend my sincere thanks to the committee members: Assoc. Prof. Dr. Somchat Jiriwibhakorn, Assoc. Prof. Dr. Keerati Chayakulkheeree, Asst. Prof. Dr. Nitikorn Junhuathon, Prof. Dr. Thanatchai Kulworawanichpong, and Asst. Prof. Dr. Tosaphol Ratniyomchai, for their valuable advice, suggestions, and support throughout the thesis evaluation process

Special appreciation is given to my parents for their enduring support, both emotionally and financially. Their belief in me has been a cornerstone of my academic journey. I am deeply grateful for everything they have done for me.

Furthermore, I would like to acknowledge all my lecturers at the School of Electrical Engineering for providing me with foundational knowledge and academic mentorship during my studies. I am also grateful to my friends, my family, and seniors, whose collaboration, encouragement, and assistance greatly contributed to this work.

Lastly, I would like to thank the faculty members and administrative staff at Suranaree University of Technology for their support and kind assistance throughout my academic life.

Nisachon Thabcha

TABLE OF CONTENTS

Page

ABSTRACT (THAI) I
ABSTRACT (ENGLISH) II
ACKNOWLEDGEMENT III
TABLE OF CONTENTS IV
LIST OF TABLES VIII
LIST OF FIGURES X

CHAPTER

I INTRODUCTION 1

1.1 General Introduction 1
1.2 Problem Statement 2
1.3 Research Objectives 3
1.4 Scope and limitations 4
1.4.1 Scope 4
1.4.2 limitations 5
1.5 Conception 5
1.6 Research Benefits 6
1.7 Structure of thesis 6

II LITERATURE REVIEW 8

2.1 Introduction 8
2.2 Literature Overview 9
2.3 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting 14
2.4 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting 18
2.5 Research Gaps in Load Forecasting Framework 22

มหาวิทยาลัยเทคโนโลยีสุรนารี

TABLE OF CONTENTS (Continued)

Page

III Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangemen 24

3.1 Introduction 24
3.2 Data Collection and Preparation 25
3.3 Feature Extraction 26
3.4 Feature Engineering 26
3.4.1 Feature Transformation / Binarization 27
3.4.2 Time-based Feature Engineering / Ordinal Time Block 27
3.4.3 Time Component Extraction 28
3.4.4 Time Series Features / Lag Features 28
3.4.5 Ambient Temperature 28
3.4.6 Dew Point 29
3.5 Advanced RNN Architectures 29
3.5.1 LSTM Architecture 29
3.5.2 Bi-LSTM Architecture 32
3.5.3 GRU Architecture 32
3.5.4 Bi-GRU Architecture 32
3.6 Workflow of the Proposed Method 33
3.7 Performance Factor 34
3.8 Simulation Results 35
3.8.1 Experimental setup 35
3.8.2 Model performance in validation and model training 37
3.8.3 Performance of each model on weekday and weekend loads ... 39

มหาวิทยาลัยเทคโนโลยีสุรนารี

TABLE OF CONTENTS (Continued)

Page

3.8.4 Performance of LSTM model on 12 test sets 44
3.9 Summary of the daily forecast 47
IV Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus 48
4.1 Introduction 48
4.2 Data Collection and Preparation 48
4.3 Feature Modeling and Representation of Electrical Load Patterns 50
4.3.1 SUT's Campus Data 50
4.3.2 New England (ISO-NE) Data 50
4.4 Feature Selection and Mutual Information 51
4.5 Modeling Approaches and Baseline Methods 52
4.5.1 Attention-LSTM Model 53
4.5.2 CNN-LSTM Model 55
4.5.3 DNN Model 56
4.5.4 Moving average 56
4.6 Time Series Cross-Validation (TSCV) 57
4.7 Shapley Additive Explanation (SHAP) 58
4.8 Forecasting Process Description 59
4.9 Load Forecasting Framework 62
4.10 Simulation Results and Discussion with SUT's Campus Data 63
4.10.1 Implementation and Experimental Setup 63
4.10.2 Mutual Information Scores of SUT Data 66
4.10.3 Daily Load Forecasting Performance 67
4.10.4 Test Set Load Forecasting Performance 71
4.10.5 SHAP-Based Feature Importance Analysis 72

มหาวิทยาลัยเทคโนโลยีสุรนารี

TABLE OF CONTENTS (Continued)

Page

4.10.6 Comparative Analysis of Feature Importance Models and Methods....73

4.11 Simulation Result and Discussion with ISO New England Data....76

4.11.1 Mutual Information Scores of ISO-NE Data....76

4.11.2 Daily Load Forecasting Performance....77

4.11.3 Test Set Load Forecasting Performance....80

4.11.4 SHAP-Based Feature Importance Analysis....81

4.11.4 Comparative Analysis of Feature Importance Models and Methods....82

4.12 Summary of Results....84

V CONCLUSION....85

5.1 Conclusion....85

5.2 Suggestions....50

REFERENCES....93

APPENDIX A Performance of LSTM Model on 12 Test Sets....**94**

APPENDIX B Implementation and Experimental Setup....**96**

APPENDIX C Test Set Load Forecasting Performance....**101**

APPENDIX D SHAP-Based Feature Importance Analysis....**107**

APPENDIX E List of publications....**113**

BIOGRAPHY....131

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF TABLES

Table **Page**

2.1 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting....10
2.2 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research....12
3.1 Performance Factor 34
3.2 Model parameter settings 36
3.3 Evaluation results of model performance in validation metrics and model training time....37
3.4 Performance of each model on weekday load 42
3.5 Performance of each model on weekend load 42
3.6 Model Performance on Validation and Average Performance on 12 Test Sets 46
4.1 Parameter Setting 64
4.2 Selecting the appropriate hyperparameter....65
4.3 Details of parameter adjustment via indicators....65
4.4 Performance of Each Model on 10 Test Sets....67

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF TABLES (Continued)

Table **Page**

4.5 Overall Test Set Performance Metrics....72
4.6 Feature Importance Ranking of NN Model by SHAP and MI....74
4.7 The best result for each model in daily forecasting....78
4.8 Overall Test Set Performance Metrics of ISO-NE Data....81
4.9 Feature Importance Ranking of ISO-NE Data....83
A.1 Performance of LSTM Model on 12 Test Sets....94

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF FIGURES

Figure **Page**

3.1 LSTM architecture 30
3.2 Bi-LSTM architecture 31
3.3 GRU architecture 32
3.4 Bi-GRU architecture 33
3.5 The Proposed Load Forecasting Workflow 33
3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models.....39
3.7 One-day-ahead electricity load forecast for each model on weekdays.....41
3.8 One-day-ahead electricity load forecast for each model on weekends.....41
3.9 Forecasting Result – Test Set 3: Exam Period.....45
3.10 Forecasting Result – Test Set 6: Start of the Semester.....45
4.1 Preparation of Actual Electricity Load Data 49
4.2 Attention-LSTM architecture 54
4.3 Expanding Window.....57
4.4 Process Description 61
4.5 The Complete Forecasting Framework 63

มหาวิทยาลัยเทคโนโลยีสุรนารี

LIST OF FIGURES (Continued)

Figure **Page**

4.6 Mutual Information Scores of SUT Data.....66
4.7 The Lowest-Performing Forecast Results in the Initial Semester Test Set.....68
4.8 The Lowest-Performing Forecast Results in the Rainy Season Test Set.....68
4.9 The Best-Performing Forecasting Results in the Test Set during the Exam Period.....69
4.10 SHAP Value of Each Model: Bi-LSTM 73
4.11 Mutual Information Scores of ISO-NE Data 77
4.12 Comparison of the best forecasting results of the CNN-LSTM model.....79
4.13 SHAP Value of Each Model of ISO-NE Data: CNN-LSTM 82
C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM 101
C.2 Performance evaluation graph of the model's test suite: a) Bi-LSTM (Top 3 Features of SHAP) and b) Bi-LSTM (Top 3 Features of MI).....103
C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features 104
D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM 107
D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM 110

CHAPTER I

INTRODUCTION

1.1 General Introduction

From the Smart Grid Driving Plan, according to the framework of the Master Plan for the Development of Thailand's Smart Grid System 2015-2036, it will promote the development of various infrastructures and the management of resources in the power distribution system that are necessary to support the transition to a new generation of power grids efficiently and environmentally friendly. This will help support Thailand to move towards clean energy and reduce carbon dioxide emissions, as well as to support the energy transition, which requires the development of a flexible power system to support the increase in renewable energy in line with the national energy plan targets. This will be an option for maintaining the country's electricity balance and managing peak electricity demand efficiently, divided into 5 main pillars, one of which is the Load Response and Energy Management System (DR & EMS). In the past few years, the demand for energy has grown rapidly. The increasing use of renewable energy and the development of digital technology have changed the operation and planning of the power system. These changes have caused both utilities and energy consumers to adopt smarter and more demand-responsive solutions to manage energy more efficiently and sustainably.

Electricity usage forecasting, which is the ability to predict future energy use, plays an important role in achieving this goal. Especially in universities or commercial buildings where electricity usage fluctuates with academic calendars, holidays, seasons, and weather conditions. Accurate short-term forecasting is essential to effectively manage the demand side, schedule, and integrate with renewable energy sources. One case is Suranaree University of Technology (SUT), located in Nakhon Ratchasima, Thailand. The university has a photovoltaic (PV) solar system and is monitored by smart meters that report electricity usage every 15 minutes. However,

the presence of solar power makes it difficult to directly observe the actual electricity demand (net load) because the measured grid load is compensated for by solar power generation. This study reconstructs the pure load by combining the grid load and solar power generation to represent the university's actual energy demand, which serves as the target for one-day load forecasting.

This research utilizes deep learning models in the field of neural networks combined with advanced methods such as K-fold validation, Mutual Information (MI), and Shapley Additive Explanation (SHAP) interpretation to improve forecasting accuracy and clarity. The goal is not only to accurately forecast electricity consumption but also to enhance the intelligence and responsiveness of the university-level energy system in line with the direction of Thailand's national smart grid system.

1.2 Problem Statement

In an era where energy demand is highly volatile and renewable energy systems are continuously integrated into the power grid, accurate electricity load forecasting is crucial for efficient energy management, especially in educational institutions like Suranaree University of Technology (SUT), which has inconsistent energy usage patterns that vary depending on weekdays, public holidays, semesters, and weather conditions such as temperature and humidity that affect air conditioning usage. The solar PV system installed within the university further complicates energy measurement because the energy meter only reports the net load after deducting the energy generated by the solar cells. This results in missing data on actual energy usage, which makes forecasting using traditional methods such as linear regression or ARIMA inaccurate and unable to handle contextual complexity.

The problem statement is as follows,

1) Accurate electricity load forecasting is essential for efficient energy management, especially in systems integrated with solar PV. However, existing models often rely on incomplete load data derived from subtracting PV generation from metered values, resulting in inaccurate representations of actual energy consumption. Additionally, traditional forecasting methods struggle to capture the complex, context-

dependent nature of energy usage patterns influenced by time, weather, and behavioral variations.

2) Improving the accuracy of electricity load forecasting requires efficient feature engineering. Because energy consumption behavior data is complex and changes with contexts such as time, weather, weekdays, or holidays, which have non-linear relationships with electricity load. However, traditional methods using linear techniques such as correlation analysis or feature selection based on statistical values often cannot capture these deep or hidden relationships. As a result, the resulting models have low performance and cannot accurately reflect energy consumption behavior in real situations or do not cover important contexts.

3) The transparency of deep learning models is also a limitation in practical applications. Although deep learning models can provide accurate predictions, they have limitations in interpreting and explaining the results. They cannot explain models with the characteristics of a black box, resulting in a lack of transparency and an inability to effectively support decision-making.

1.3 Research Objectives

To address such challenges, this study aims to develop a novel short-term electricity load forecasting model using deep learning techniques in the RNN family and enhance its accuracy with contextual feature engineering and interpretability tools to enable more accurate, transparent, and efficient energy decision-making for short-term (one-day-ahead) electricity load forecasting.

The main objectives of this thesis are as follows,

1) To develop a one-day electricity load forecasting model using deep learning techniques. We will compare deep learning models with basic models, including Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. Specifically, we will examine the model with the highest accuracy based on real-world load data reconstructed from electricity meters and solar panels.

2) To improve accuracy and reliability through the design of appropriate and high-quality features. Mutual Information (MI) is a promising approach for capturing

complex nonlinear energy consumption patterns. and systematic evaluation using performance indicators including RMSE, MAE, and MAPE.

3) To improve the transparency and interpretability of the model, SHAP analysis is used to identify influential features and analyze their importance to gain a deeper understanding of the model's performance.

1.4 Scope and limitations

1.4.1 Scope

This study focuses on electrical load forecasting using deep learning techniques. In addition, it studies the use of feature engineering techniques to improve the input data for the learning model, including the selection of necessary features and the description of variables through the model. In summary, the scopes are considered as follows:

1) This study focuses on the development of a one-day-ahead electricity load forecasting model for Suranaree University of Technology (SUT) using deep learning techniques including moving average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM.

2) The target variable used in the forecasting model is the pure load, reconstructed by combining grid-based electricity consumption with solar PV generation data to represent the university's actual energy demand.

3) The input features are designed to reflect real-world contexts, including temporal data (e.g., yearly hours, quarter-hour intervals), calendar effects (e.g., weekdays, holidays, long weekends), environmental factors (e.g., ambient temperature, relative humidity), and lag-based load patterns (e.g., previous day's load).

4) Model training utilizes K-Fold cross-validation, and performance is evaluated using standard forecasting metrics: RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error).

5) The study incorporates SHAP (Shapley Additive Explanations) to interpret model outputs, allowing for an understanding of feature importance and offering insights for practical improvements in energy management strategies.

1.4.2 Limitations

The limitations of this study are to consider three case studies as follows:

1) While the pure load is reconstructed by combining grid and PV data, minor discrepancies may exist due to unaccounted factors such as system losses or unmonitored sub-loads.

2) SHAP analysis for RNN-based models provides approximate feature importance and may not fully capture all intricate dependencies within deep neural networks.

1.5 Conception

The main idea of this research comes from the need for accurate and interpretable electricity load forecasting, which is essential for planning and managing energy systems in an era of rapid and uncertain energy usage patterns, especially in the context of complex power systems that require transparent, data-backed decision-making.

This research proposes a forecasting structure for electricity consumption using time-sequence deep learning models, including LSTM, GRU, Bi-LSTM, and Bi-GRU. Moving Average, DNN, CNN-LSTM, and Attention-LSTM are compared to obtain comprehensive results. Electricity consumption data is organized using Day-Aware Data Arrangement to clearly distinguish electricity usage patterns. Feature engineering is performed to identify influential variables, and model robustness is evaluated using twelve carefully designed test sets representing different times and operating conditions. To increase the reliability and understanding of the model, the system applies SHAP (Shapley Additive Explanations) to explain the model results and uses Mutual Information (MI) to select non-linear features, as well as Time Series Cross-Validation (TSCV) to evaluate the results comprehensively according to the context of time-sequence data, covering accuracy indicators of RMSE, MAE, and MAPE to evaluate the model's capabilities in multiple dimensions.

The integration of these components aims to create an electricity load forecasting system that is not only highly accurate but also whose results can still be interpreted well, are reliable, and can be confidently applied in real situations.

1.6 Research Benefits

This research presents a practical and interpretable deep learning-based framework for one-day-ahead electricity load forecasting, using real-world data from a university campus with integrated solar PV systems. By reconstructing the pure load and incorporating contextual features such as weather, time, and holidays, the model better reflects actual electricity demand and addresses forecasting challenges in renewable-integrated environments.

At the institutional level, the model supports more accurate energy planning, demand-side management, and cost optimization. At a broader level, it serves as a scalable solution for similar commercial or academic settings, promoting efficient energy use and integration of renewable energy. At the national level, the study aligns with the goals of energy transition and smart grid development by enabling more flexible, data-driven, and responsive electricity systems that can support peak demand management and sustainability targets.

By combining forecasting accuracy with explainability through SHAP, the research contributes not only to academic advancement but also to the foundation of smarter and more sustainable energy infrastructure.

1.7 Structure of thesis

This thesis is organized into five chapters, outlined as follows:

Chapter I: This chapter presents the background and significance of the thesis, including the challenges arising from complex variations in electrical load patterns. It discusses the problem, research objectives, scope and limitations, the proposed methodology concept, and the overall structure of the thesis.

Chapter II: This chapter covers theoretical principles and research on electrical load forecasting models, including a review of relevant optimization methods, feature analysis, and gap analysis in existing research.

Chapter III: This chapter presents one-day forecasting of electricity consumption using supervised deep learning techniques combined with daily data sorting to select models most likely to be further developed for maximum accuracy and to find the most efficient data management methods for achieving optimal results.

Chapter IV: This chapter discusses an advanced data-driven load forecasting framework for high-volume solar power systems: a case study of the SUT campus and data from ISO New England, demonstrating data preparation, model selection, simulation, and feature analysis using various tools for forecasting accuracy and reliability.

Chapter V: The final chapter summarizes the key research findings and evaluates the effectiveness of the proposed methodology, including a comparison of the results of the models and feature analysis tools from simulation and analysis results.

มหาวิทยาลัยเทคโนโลยีสุรนารี

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Load forecasting is the heart of power system management in the digital age, especially in the context of smart grids that require short-term and long-term forecasting accuracy to increase efficiency in planning energy production and distribution with stability and efficiency. Due to the complexity of time series data that is nonlinear and has various environmental factors that affect electricity loads, such as weather, consumer behavior, working days or holidays, etc., it is necessary to develop analysis techniques that can handle such data appropriately. In recent years, deep learning techniques have been increasingly popular for load forecasting, especially sequence models. However, the accuracy of forecasting models does not only depend on the model structure but also on appropriate feature selection and feature engineering that allow the model to learn data more deeply. In addition, model explainability is another important issue, especially in systems that require transparency and reliability to meet the requirements of accuracy, explainability, and appropriate feature selection. This research is interested in the approach that focuses on developing an explainable time series model by combining contextual feature selection and temporal cross-validation to obtain reliable results in the context of real load forecasting. The following topics will explore the knowledge related to various research areas, including data preparation and feature selection techniques, the development of deep learning time series models, model description techniques, and comprehensive model evaluation approaches in load forecasting.

2.2 Literature Overview

To provide an overview of past approaches to load forecasting using deep learning techniques, this research has compiled and compared related research, divided into 2 groups: the research group using deep learning models for load forecasting and evaluating model performance and the research group focusing on feature selection and creation and model explainability methods. Table 2.1 summarizes the main components of each research, including feature engineering, selecting the best model from forecasts compared to other models or categories, metrics used for evaluation, techniques used for feature selection, and model explainability.

From the analysis of the research in Table 2.1, many works using various deep learning models in the context of load forecasting can teach temporal relationship and non-linear behavior better, along with the selection and design of specific features in contextual feature engineering. However, there is still a lack of non-linear feature selection, evaluation by time-based data partitioning, and the clear description of the model results. Therefore, our research aims to develop an approach that combines the selection of non-linear features using MI, time-based cross-validation for time series data, and model description using SHAP for deep analysis under the framework of LSTM, GRU, Bi-LSTM, and Bi-GRU models to create an accurate and easy-to-understand load forecasting framework.

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.1 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting

Ref	Feature Engineering	Best Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Panapongpakorn & Banjerdpongchai, 2019	✓	LSTM	✓	✓	-	MI	-
Vanitha & Swaroopan, 2023	✓	Bi-LSTM	✓			Correlation	-
Mpawenimana et al., 2020	STDAN	LSTM	-	✓	✓	Active Power	-
Prakash et al., 2023	✓	LSTM, Bi-LSTM	✓	✓	✓	-	-
Habbak et al., 2023	✓	NN	✓	-	✓	-	-
Atef & Eltawil, 2020	✓	Bi-LSTM	✓	✓	✓	-	-
Pavlatos et al., 2023	✓	Bi-LSTM	✓	-	✓	-	-

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.1 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting (Continued)

Ref	Feature Engineering	Best Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Sias et al., 2024	✓	Multi Bi-GRU	-	✓	✓	-	-
Abumohsen et al., 2023	✓	GRU	✓	✓	✓	-	-
Cordeiro-Costas et al., 2023	✓	LSTM	Normalized	✓	-	-	-

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.2 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research

Ref	Feature Engineering	Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Lee et al., 2023	✓	XG Boost	-	-	✓	-	SHAP value based
Neubauer et al., 2025	✓	ED, XG Boost	Normalized	✓	-	-	SHAP, Attention
van Zyl et al., 2023	✓	CNN	✓	-	✓	MI	SHAP, Grad-CAM
Jang et al., 2023	✓	RAID	-	-	✓	-	SHAP
Koprinska et al., 2015	✓	LR, BPNN	✓	-	-	MI, Relief	-
Pirbazari et al., 2019	✓	GBRT	-	✓	-	MI, RFE, Elastic Net, F-regression	-
Liu et al., 2024	✓	KAN, Transformer, Informer	✓	-	-	Correlation Feature Self-Assignment Module	KAN

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table 2.2 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research (Continued)

Ref	Feature Engineering	Model	RMSE	MAE	MAPE	Feature Selection	Model Explainability
Maarif et al., 2023	✓	SL-LSTM, DL-LSTM, Bi-LSTM	✓	-	-	SHAP	SHAP
Wang et al., 2025	✓	SMLR, XG Boost, LASSO-XG Boost	✓	✓	✓	SHAP	SHAP
Van Zyl et al., 2024	✓	1D-CNN	✓	✓	✓	SHAP, XG Boost, Feature Importance, Variance Threshold, NSGA-II	SHAP, Grad-CAM
Wu et al., 2020	✓	Attention-LSTM	✓	✓	-	Attention weights	Attention weights
Ullah et al., 2024	✓	CNN-LSTM	✓	✓	✓	Correlation	-
Fan and Hyndman, 2012	✓	Semi-parametric additive	✓	✓	-	-	-
Shi et al., 2018	✓	Deep RNN	✓	✓	-	-	-

มหาวิทยาลัยเทคโนโลยีสุรนารี

From Table 2.1, additional research related to the introduction of feature selection techniques and model description capability in load forecasting is summarized in Table 2.2, which summarizes the main components of each research study, including feature engineering, all models used in forecasting, evaluation metrics, feature selection techniques, and model description capability. The trend of developing load forecasting models focuses on interpretability and accurate feature selection to increase both efficiency and transparency. Many research studies focus on using SHAP and other XAI techniques to analyze feature importance. However, there are still gaps in using filter-based methods such as MI and temporal cross-validation, including using SHAP for feature selection, which our research will supplement to obtain a system that is both accurate, interpretable, and reliable in the long run.

2.3 Overview of Feature Techniques, Model Selection, Evaluation Techniques, and Feature Selection in Electrical Load Forecasting

To delve deeper into the trends, technologies, and limitations of the related research in Table 2.1, this section describes in-depth details of each selected research, focusing on feature selection methods, techniques used for load forecasting, proposed models, evaluation metrics, and limitations of each work, and proposes ways in which we can improve upon those limitations within the framework of load forecasting with deep learning.

(Panapongpakorn & Banjerdpongchai, 2019) This research presents a non-linear feature selection (Mutual Information—MI) method, where the information relationship between two variables captures the non-linear relationship to select the important input variables for the model. This research aims to apply autoregressive integrated moving average (ARIMA) and seasonal autoregressive integrated moving average (SARIMA), artificial neural network (ANN), and long-short term memory (LSTM)-based recurrent neural network (RNN) to improve the performance of short-term electrical load forecasting (STLF) in energy management systems (EMS). Time series features were structured, and the performance was measured with Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The models from neural networks have

better performance than the models from time series analysis, with the best performing model being LSTM. However, the performance does not cover the division of train/validation sets by time, and there is no time-aware dataset division (temporal cross-validation), so the model still cannot learn well in all time periods. The technique used to describe the model, which features affect the model output, has not been mentioned, and there is still a lack of in-depth analysis of the specific model directly.

(Vanitha & Swaroopan, 2023) This paper proposes a method to compare Stacked GRU-RNN and Bidirectional LSTM (Bi-LSTM) artificial neural network models in forecasting electricity load and renewable energy production in Smart Grid systems using real energy consumption data along with the analysis of the relationship of the input data to select the most influential features from the use of correlation analysis. The feature selection is linear relationship statistics. The linear relationship between 2 variables has a point that it does not support non-linear relationships from some features, so it does not cover all the impacts of the features on the model output. The results show that the Bi-LSTM model provides high accuracy in energy and load forecasting.

(Mpawenimana et al., 2020) This paper compares the short-term and medium-term energy load forecasting capabilities of LSTM and ARIMA models in the context of a smart home with an energy harvesting system. A new data preparation technique called STDAN (Same Time a Day Ago or Next) is proposed for handling missing values, which performs better than the average or previous value filling method. The paper states that the prediction uses only active power data from the dataset, which can be considered a feature selection, focusing on a single feature believed to be related to the energy load, which is a manual selection. The experiments show that LSTM significantly outperforms ARIMA in all cases. More sophisticated feature engineering can also be developed to help the model learn better. There is no mention of efficient feature selection and no explanation of which features influence the results.

(Prakash et al., 2023) This paper presents a forecast of the daily COVID-19 cases in India over the next 6 months using time series analysis techniques such as machine learning and deep learning, as well as a hybrid model. To compare the performance of different models in predicting the number of infected people using historical data

and evaluating the performance through statistical indicators. The experimental results show that basic models such as Prophet, Vanilla LSTM, and Bi-LSTM give more accurate results than complex or hybrid models. This research is not directly in the context of energy or electrical load, and there is no feature selection or model description.

(Habbak et al., 2023) This article is a systematic review covering load forecasting techniques in smart grids. The analysis results show that techniques using Machine Learning (ML) and Neural Networks (NN) provide the highest forecasting accuracy, with significantly lower Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) values than other methods. No new methods or in-depth comparisons were found, efficient feature selection, and lack of approaches to explain causal models.

(Atef & Eltawil, 2020) This article presents a comparative study between unidirectional (uni-LSTM) and bidirectional (bi-LSTM) LSTM models in 1, 2, and 3-layer nesting, using an automatic parameter tuning tool to find the best model. The experimental results show that Bi-LSTM significantly outperforms Uni-LSTM and Support Vector Regression (SVR) in terms of forecasting accuracy. When considering error values such as RMSE, MAPE, and MAE, deep nested models do not significantly outperform single-layer models, and they take longer to compute. There is no mention of efficient feature selection, and there is no explanation of how each feature affects the other.

(Pavlatos et al., 2023) This paper proposes a Bi-LSTM based neural network-based electrical load forecasting framework that can simultaneously process historical and future data to improve the forecast accuracy of temporal data. Experiments on multiple datasets show that Bi-LSTM significantly outperforms RNN, LSTM, and GRU. MAE and RMSE are used for performance evaluation. It is found that additional performance evaluation techniques can be added to cover more comprehensive models. There is still no efficient feature selection, and there is a lack of approaches to explain the causal model.

(Sias et al., 2024) This research presents a multivariate bidirectional GRU model combined with periodic features for forecasting energy loads and renewable energy sources (e.g., solar cells). The model uses multivariate data, such as system energy

data and weather data (for power generation), and tests are conducted on daily, hourly, and minute data levels. The experimental results show that the proposed model has the highest accuracy. And it takes less processing time than the basic Bi-GRU and Bi-LSTM models. From this research, it is not clear which features affect the model performance, and it lacks the step of explaining the relationship of the features to the results.

(Abumohsen et al., 2023) This paper uses deep learning techniques to forecast electrical power loads: Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNN). By testing all three models, it was found that the GRU model gave the best results in terms of accuracy and the lowest error. There are still limitations found in this research: the lack of feature selection steps and model result explanations, and the analysis only analyzes the results, not the components of the model. It is possible to analyze the behavior of the GRU in more detail.

(Cordeiro-Costas et al., 2023) This paper focuses on evaluating and comparing the accuracy of various artificial intelligence (AI) techniques in forecasting electrical loads in buildings. To compare with the forecasting of loads in single-family homes in the United States, using a 10-fold cross-validation technique to assess the bias and variance of each model. The experimental results found that LSTM is the model with the lowest error distribution in the validation and test datasets. Using cross-validation General models may not be suitable for time series data and lack in-depth analysis of feature selection.

From the research studies in the past few years, there is a clear trend in applying deep learning techniques together with the selection and design of specific features, or so-called contextual feature engineering, to address the complexity of electrical load data in real contexts. Traditional models such as ARIMA or linear regression often cannot effectively handle behaviors that change with context, such as days of the week, weather conditions, or holidays. While deep models such as LSTM and GRU can learn temporal relationships and non-linear behaviors better. However, these models are most effective only when they receive input data with appropriately designed features, such as adding calendar effects, temperature, humidity, sunlight, and various time indicators such as rush hours or public holidays, which allow the

model to learn complex patterns of electrical loads more accurately. Therefore, research in this direction is continuously increasing, focusing on both developing model architectures and enhancing the potential of input data.

2.4 Comprehensive Overview of Modeling, Feature Engineering, Evaluation, Feature Selection, and Explainability in Load Forecasting Research

To address the challenge of developing a load forecasting model that is not only accurate but also transparent and interpretable, several recent studies have focused on appropriate feature selection, comprehensive evaluation, and the use of Explainable AI (XAI) techniques to enhance the reliability of the model. This chapter presents an in-depth analysis of those studies, examining the key elements of the model, feature selection, evaluation, and model explanation, and points out the limitations and solutions that will be further developed in our work. From Table 2.2, it can be explained as follows:

(Lee et al., 2023) Using Explainable AI for Transparency in Load Forecasting, we propose a new metric that makes the feature importance analysis more reliable, supporting applications in real systems that require transparency, such as energy systems. Since the work has not been tested with multiple datasets or models, which may limit the generality of the metric.

(Neubauer et al., 2025) This paper presents a multi-step heating load forecasting model that emphasizes the ability to explain AI using SHAP values and temporal attention mechanisms to enhance the understanding of the process of data-driven forecasting models. Comparison with other explainable techniques such as MI is not discussed.

(van Zyl et al., 2023) This article analyzes and compares the capabilities of two types of Explainable Artificial Intelligence (XAI) techniques: Grad-CAM (Gradient-weighted Class Activation Mapping) and SHAP (Shapley Additive Explanations) in performing feature selection for energy demand forecasting models from large time series using a multi-headed CNN as the model base. CNN models may not be suitable

for time series load data with clear temporal dependency and lack of comparison with other models for comprehensive results.

(Jang et al., 2023) This article presents a Robust and Interpretable DPLF (RAID) model for Daily Peak Load Forecasting (DPLF) that is both accurate and interpretable using multiple Deep Neural Networks (DNNs) with SHAP values analysis to explain the importance of features. However, there is no evaluation of temporal generalization or cross-time validation.

(Koprinska et al., 2015) This paper presents a two-step feature selection method for 5-minute electricity load forecasting. Initially, features are selected based on daily and weekly electricity usage patterns, such as load from the previous hour, and feature selection techniques such as Relief and Mutual Information are used to assess the importance of features and select the most suitable feature set. The results show that using this feature selection method can reduce the number of features without compromising the forecast accuracy. However, the work has not included the interpretability perspective from XAI or analyzed the impact of deep features.

(Pirbazari et al., 2019) studied the effectiveness of feature selection methods in short-term household energy forecasting, especially in the context of smart meter data. It shows that feature selection techniques can effectively screen out suitable variables specific to each household, speed up model training, reduce model complexity, and produce forecasting results that are close to or better than those without feature selection. This research lacks the mention of deep model description techniques.

(Liu et al., 2024) This paper proposes a new model called Correlation Feature Self- Assigned to flexibly combine features based on the degree of data correlation, the Interpretable Kolmogorov-Arnold Network (KAN) is the ultimate predictor for forecasting various types of energy loads in IES. There is no comparison with other XAI techniques such as SHAP or MI, which are suitable for non-linear data.

(Maarif et al., 2023) This research successfully developed an energy forecasting model using LSTM combined with XAI, which is not only efficient in forecasting but also clearly explains the model results, allowing users to understand and trust the results obtained from the model. The Bi-LSTM and double-layer LSTM models perform

slightly better than single-layer LSTM, with lower RMSE. This research uses SHAP to select important features and train a new model, which significantly improves forecasting performance. However, there is no mention of feature selection before model training. Immediately introducing all features into the model may cause overfitting or noise from unimportant features. The work does not mention the use of techniques such as k-fold cross-validation or other data partitioning to prevent overfitting. A single train/test partition may result in model bias or incorrect performance evaluation in real-world use cases.

(Wang et al., 2025) This research demonstrates the effectiveness of using the SHAP technique to analyze the importance of features in a short-term load forecasting model by identifying the features that have the most impact on the model results, which helps to improve the model accuracy and reduce the complexity of the model by removing unnecessary features. Using the SHAP technique also increases the transparency of the model operation, allowing users to understand and verify the model operation effectively, which is very useful for applying the model in real situations that require reliability and transparency. The research found that the model description from SHAP has not been linked to a systematic feature selection strategy.

(Van Zyl et al., 2024) This article presents the ability of SHAP to reduce the number of features without significantly degrading the model performance. In some cases, it helps to make the model more stable and improve the interpretation of the results. However, it is not suitable as a method for selecting “initial” features but is suitable for fine-tuning after obtaining the initial feature set. Although SHAP is not a direct feature selection technique, the researchers use SHAP scores to rank the features and then build a new model by comparing it with other feature selection methods. Therefore, SHAP is used strategically to “select features by importance.” Although this work uses a variety of feature selection methods, it still lacks a filter method that supports non-linear relationships. Independent of the model, such as MI, the addition of MI enhances the "pure data" view that is independent of the structure or bias of any model. In addition, a single model (1D-CNN) is used as the basis for analysis. The results of SHAP depend on the model used. If other models are used for comparison, such as LSTM or Bi-LSTM, the ranking of features may be different. Using only one model may

limit the reliability of the conclusion of "which features are important" after model training. And cross-validation is used to evaluate the accuracy of the model for each selected feature set.

(Wu et al., 2020) This paper proposes a combined deep learning framework based on an attention-based LSTM model to improve short-term forecasting performance by selectively focusing on important temporal features. The attention mechanism enhances the model's ability to capture key dependencies in time-series data and improves prediction accuracy. However, the study does not perform explicit feature selection, and the explainability of the model is limited to attention weights without providing a comprehensive feature-level interpretation such as SHAP.

(Ullah et al., 2024) This paper presents a comprehensive review and simulation study of short-term load forecasting using CNN–LSTM hybrid models, highlighting their ability to extract spatial features via CNN and temporal dependencies via LSTM. The results demonstrate improved forecasting accuracy compared to single models. However, the study mainly focuses on performance evaluation and lacks feature selection analysis and model explainability techniques to interpret how input features influence the forecasting outcomes.

(Fan & Hyndman, 2012) This paper proposes a semi-parametric additive model for short-term electrical load forecasting by decomposing load patterns into trend, seasonal, and smoothing components using statistical feature engineering techniques. The model provides good interpretability and stable forecasting performance; however, it is limited in capturing complex nonlinear relationships and does not incorporate feature selection or advanced model explainability methods.

(Shi et al., 2018) This paper proposes a deep learning-based load forecasting framework using a novel pooling deep recurrent neural network to capture complex temporal dependencies in household electricity consumption. The proposed model significantly improves forecasting accuracy compared to traditional methods; however, it treats the model as a black box and does not provide explicit feature importance analysis, feature selection, or explainability to support model interpretability and reliability.

While deep learning models have become increasingly popular due to their ability to produce highly accurate results, a major problem is the interpretability of the models, which often have a black box appearance. That is, even if the results are accurate, they cannot clearly explain which factors the model uses to make decisions. In the context of energy management in institutions or organizations, accurate forecasting alone may not be enough if decision-makers cannot understand or trust the decision-making process of the model. Therefore, modern research has begun to focus on interpreting model results using techniques such as Shapley Additive Explanations (SHAP), which can indicate how much each feature influences the model's results. This technique allows for a transparent analysis of which factors drive the forecast results, such as temperature, time of day, or day of the week, and can be used to further develop energy policies or improve models in the future. Combining the ability to explain results with forecast accuracy is an important direction in the development of AI models for sustainable energy management.

2.5 Research Gaps in Load Forecasting Framework

From the review of past research, it was found that there is a tendency to use various deep learning models, such as LSTM, Bi-LSTM, and CNN, in the context of energy load forecasting, both short-term and long-term, which allows the model to learn the temporal characteristics of the data efficiently. However, many researchers still lack feature selection in a way that supports non-linear relationships, including model evaluation in a way that is appropriate for temporal data, such as time-aware data partitioning or temporal cross-validation. In addition, although many researchers have started to use Explainable AI (XAI) techniques such as SHAP and Grad-CAM to analyze the importance of features and increase the transparency of the model results, there are still significant gaps in the application together with filter-based feature selection methods such as Mutual Information (MI), which can help filter important features directly from the data, independent of the model, and reduce bias from the model structure itself. In evaluating the forecast results, most researchers use various indicators to measure the performance of the model in many dimensions, including the accuracy of the prediction value, the average absolute error, and the relative error

to the actual value. Using these indicators together allows for a more comprehensive analysis of the quality of the model, especially when applied in real situations that require both accuracy and reliability.

Based on the above findings, this research aims to develop a comprehensive load forecasting approach encompassing three main dimensions: feature selection using mutual information to support nonlinear relationships; modeling results through Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models for multi-dimensional model coverage; detailed model description within the NN group using SHAP; evaluation using Temporal Cross-validation suitable for timeline data; feature development and enhancement using Feature Engineering to suit data characteristics; and the use of multi-dimensional metrics including RMSE, MAE, and MAPE. This will result in a developed forecasting system that is accurate, transparent, and interpretable, suitable for use in smart energy systems requiring long-term reliability.

มหาวิทยาลัยเทคโนโลยีสุรนารี

CHAPTER III

Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement

3.1 Introduction

In the context of electrical load forecasting, choosing the right method for the highly complex data is an important issue, especially for electrical load data that has a non-linear relationship and changes significantly over time, such as weekdays, holidays, or school openings and closings, which all affect energy use behavior at the institutional or large organization level, such as universities.

Current research trends have begun to focus on the use of deep learning models, especially the RNN family of models such as LSTM, GRU, Bi-LSTM, and Bi-GRU, which are designed to efficiently handle time series data. The strength of these models is their ability to learn patterns of past data to predict the future. However, these models often have limitations in terms of transparency in explaining their results, which raises questions about their practical reliability. To solve this problem, research focuses on the process of selecting appropriate features, adding contextual information such as time, weather, calendar, and holidays, and using model testing and evaluation techniques that are consistent with time series data, such as data partitioning by the Expanding Window method, which does not leak future data to model training. In addition, the use of the SHAP technique allows for a clearer explanation of model decisions, both at the individual data point level and in general.

This chapter aims to describe the research process used to build and test the electrical load forecasting model, which will serve as an important basis for the comparative performance analysis in the next chapter. Parts of this thesis have been published in the proceedings of the 2025 13th International Electrical Engineering Congress (iEECON2025), is shown in the Appendix D.

3.2 Data Collection and Preparation

The datasets collected and used in the forecasting include electric load data from utility meters, which shows historical data from the Automatic Meter Reading (AMR) system maintained by the Provincial Electricity Authority (PEA), which records electricity usage according to load demand at the university, as well as solar energy production data, which records energy production from meters installed in the area with solar panels at Suranaree University of Technology in Thailand. Data regarding compensatory holiday long weekends is gathered from university announcements and the academic calendar. A reliable public website with an API for direct data extraction is the source of historical condition data. Every 15 minutes for a year, from January to December 2023, these statistics are collected and compiled. The electrical load forecasting utilizes 35,040 datasets in total. In the data cleaning process, which is an important process in preparing various data before using it for forecasting in the model, the following methods are used: Handling missing values (Handling Missing Values) will delete the missing data range or the data range with abnormalities and add new data from the method of calculating the average and transforming the data format to be appropriate (Data Formatting) using Timestamp to help create the Time Factor and Social Factors features, helping the model to learn the relationship of the data better and be more efficient in forecasting. Since the electrical load in the university is a high PV penetration load with a proportion of solar energy produced and sent to the total power system, the electric load data from utility meters or net load obtained is data that is mixed with solar energy production data or PV generation, making it possible to see electricity usage according to the load demand that is unrealistically low. Therefore, PV Generation is added back to obtain pure load or electrical load data that can be used to assess the actual electricity demand in forecasting.

The data is divided by taking the entire cleaned dataset for the year and splitting it into three sets including the training set, which uses 80% of the total data; the validation set, which uses 20% of the data for parameter tuning; and the test set, which consists of a separate portion of data that the model has never seen before, reserved for final evaluation.

3.3 Feature Extraction

Assigning features to existing data is very important in forecasting electrical load. Because features are variables that will be fed into the model to help the model learn relationships and generate accurate forecast results. Features used in forecasting this type of data are a collection of parameters that are significant to load forecasting and are created by considering the context of time and day. Creating features from data can be divided into two types as follows: The first type is creating features that are Day-Aware Data Arrangement, which is a method of arranging data that takes into account the context of time and day to allow the model to learn the characteristics of the data on each day or specific time period. It consists of Hour of Day, which is a value that specifies the hours in one year; Quarter-Hourly Interval, which is a value that specifies the minutes every 15 minutes in each hour; Weekdays and Weekend, which are values used to separate data between weekdays and weekends; Daytime and Nighttime, which are values used to separate data between day and night; Compensatory Holiday, which is a value that specifies whether that day is a holiday or not, Long Weekend, which is a value that specifies that period is a special holiday or festival. The second type is a feature that is weather data, consisting of ambient temperature data and relative humidity data. The number of data sets used to create and divide the feature data sets will be the same as the data used in forecasting electrical load.

3.4 Feature Engineering

Feature engineering is an important step in developing a load prediction model, playing a role in creating appropriate variables that accurately represent temporal behavior, which will improve both the accuracy and explainability of the model. From the collection and review of research, feature engineering techniques will be used. In this research, the techniques will be divided into two main groups: feature generation and external features. The first group is featuring generation, which is the creation of new features from existing temporal data and the transformation of temporal data that may not be easily understood by the model into logical information that is meaningful according to the context. as described as follows.

3.4.1 Feature Transformation / Binarization

Variable transformation and binarization are the processes of transforming the source data into a format that is easy to interpret or respond to the model, as follows: Create a feature that divides Monday to Friday to represent school days and Saturday and Sunday to represent weekends, with the numbers 0 and 1, so that the model clearly understands the difference between Monday and Sunday. Create a feature that divides day and nighttime with the numbers 0 and 1, such as 8:00 - 17:00, which is daytime and is 1, and the rest is nighttime, which is 0. Dividing day and nighttime into categories that are easier to understand. Creating a feature that does not just divide normal holidays and school days but can be adjusted according to special events, such as compensatory holidays or school days, helps the model better understand the behavior at each time period.

3.4.2 Time-based Feature Engineering / Ordinal Time Block

Segmenting data into meaningful timelines with 1 year of data every 15 minutes creates a feature that allows the model to learn the day and time, such as day 1, where 00:00 - 00:45 will be set as 0, time 01:00 - 01:45 will be set as 1, time 02:00 - 02:45 will be set as 2, and so on until 23:00 - 23:45 will be set as 23, considered day 1, and day 2 is set as the same. The method and advantages of this feature are as follows:

Ordinal Encoding: Use sequential numbers (0, 1, 2, ... 23) to represent time of day, which helps the model to recognize the continuity of the data.

Time Segmentation: Divide time periods into groups with different usage patterns, such as morning (Morning), afternoon (Afternoon), evening (Evening), and night (Night).

Pattern Recognition: The model can learn the trends of each time period, such as energy consumption may be higher in the morning or lower at night.

Use this feature to forecast energy usage by letting the model understand that different days can have different energy usage patterns, helping the model better understand the context of that time period.

3.4.3 Time Component Extraction

Decomposing time components into separate features creates a new feature that describes each sub-hour: 00.00 is 0, 00.15 is 15, 00.30 is 30, and 00.45 is 45. After 01.00, it is 0; 01.15 is 15; 01.30 is 30; and 01.45 is 45. After 02.00, it is 0. This continues so that the model can capture the relationship between time periods and different load behaviors. The model can understand the recurring behaviors during each sub-hour, and this helps to understand the trends in energy usage over time. For example, every hour from 0-45 may have a similar energy usage pattern.

3.4.4 Time Series Features / Lag Features

Building features from previous values (lag) is an important basic technique in the time series data analysis. It uses historical load values to build features such as load in the same period of the previous day (Load t-1d). This method allows the model to learn trends and seasonality efficiently.

In the second group there are external (exogenous) features, external variables that do not arise from the process of the target variable itself but have an impact or relationship with the forecast of the target variable and can be used as additional input to the forecasting model to increase its accuracy, consisting of the following techniques:

3.4.5 Ambient Temperature

Ambient temperature is an external factor that directly affects energy use behavior, such as air conditioning or heating use. This information is often used in conjunction with load data to help models understand the physical relationship between weather conditions and energy demand.

3.4.6 Dew Point

The Dew point is a measure of humidity in the air, which affects temperature perception and the use of cooling or heating equipment, especially in regions with high humidity. Incorporating dew point values into models allows for a more accurate capture of energy consumption behavior.

3.5 Advanced RNN Architectures

The design of the model architecture is the heart of developing the electrical load forecasting model. Due to the nature of electrical load data being a time series with significant variations over time and period, selecting a model that can accurately capture the characteristics of the data sequence is essential. In this section, we have selected the architecture of the Recurrent Neural Networks (RNNs) model that has been improved to better remember long-term data, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), as well as the Bidirectional version (Bi-LSTM and Bi-GRU), which can see the context from both the past and the future in the time sequence, allowing a deeper understanding of the relationships in the data.

3.5.1 LSTM Architecture

The Long Short-Term Memory (LSTM) model is a structure of Recurrent Neural Network (RNN) designed to solve problems and manipulate time-series data. It consists of a Forget Gate that determines which data should be forgotten from the original cell state. Let x_t be the input feature vector at time t, $\sigma(\cdot)$ be the sigmoid activation function, h_{t-1} be the hidden state from time $t-1$, and b_f be the bias vector of the forget gate, W_f be the learnable weight matrix of the forget gate. The forget-gate output is computed as shown in Equations (3.1).

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \tag{3.1}$$

where $f_t \in [0,1]$ is close to 1, indicating retention, while values close to 0 indicate erasure. This helps determine whether data should be retained. The next component, the Input Gate (i_t) and the Candidate Cell State, decides which new data to add to the cell state, as shown in Equations (3.2) and Equations (3.3).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\tilde{C}_t = \sigma(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.3)$$

where i_t controls which new data should be imported, and $\tilde{C}_t$ represents the new data processed (candidate memory). The next component, the Update Cell State, combines the Forget Gate and the Input Gate to create a new cell state, as shown in Equations (3.4).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.4)$$

where the forgotten data is multiplied by f_t, and the new, important data is multiplied by i_t and added together. The final section is the Output Gate (o_t) and Hidden State. These define the data to output as hidden states for future forecasting, as shown in Equations (3.5) and Equations (3.6).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.5)$$

$$h_t = o_t * tanh(C_t) \quad (3.6)$$

where h_t is the hidden state, used to predict the electrical load at the next point.

LSTM is a special kind of RNN that can identify long-term dependencies. LSTMs are purposefully designed to avoid the problem of long-term reliance. They essentially behave in such a way that their default state is long-term memory retention. For time series forecasting, they are hence the perfect parallel. Fig. 3.1 describes how LSTM cells are represented graphically.

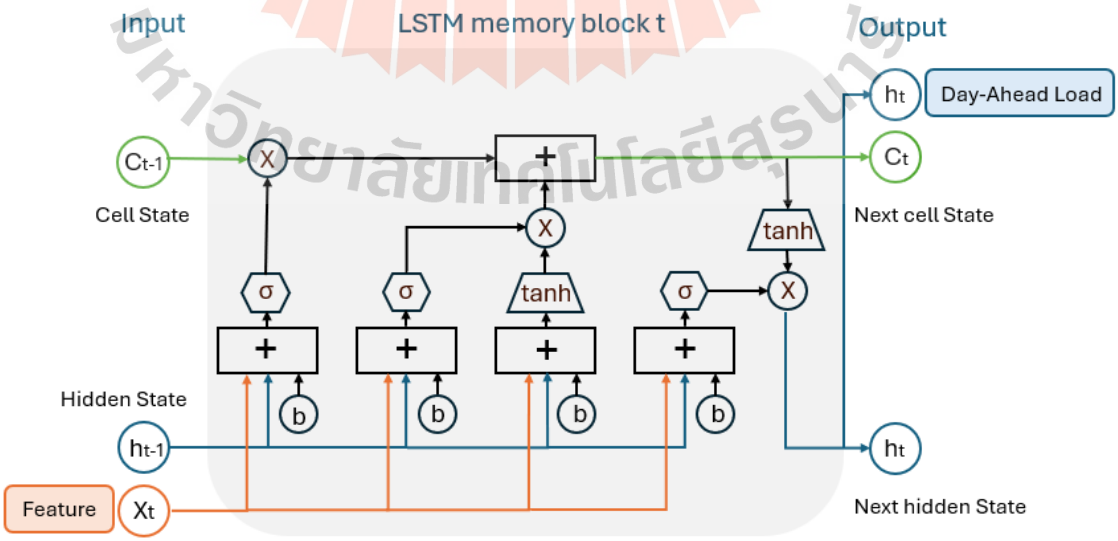


Figure 3.1 LSTM architecture

3.5.2 Bi-LSTM Architecture

The Bidirectional Long Short-Term Memory (Bi-LSTM) model is a structure that is developed from LSTM to increase the ability to learn sequences of data that have relationships in both the past (past dependencies) and the future (future dependencies) simultaneously. It consists of two LSTM layers that work together: a forward LSTM that processes data from the past to the present and a backward LSTM that processes data from the future back to the present at any time t. The results of both directions are combined into a single hidden state as shown in Equations (3.7).

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \tag{3.7}$$

where $\vec{h}_t$ is the hidden state from the Forward LSTM, $\overleftarrow{h}_t$ is the hidden state from the Backward LSTM, and h_t is the combination of the two results.

Bi-LSTM is a sequence processing model that is made up of two LSTMs, one of which accepts input in a forward direction and the other in a backward manner. Bi-LSTMs give the network access to additional information, which improves the context of the algorithm. The primary advantage is that each input sequence piece includes both current and historical data. Therefore, Bi-LSTM can produce a more pertinent outcome by merging LSTM layers from both directions. The configuration of LSTM layers in Bi-LSTM is shown in Fig. 3.2.

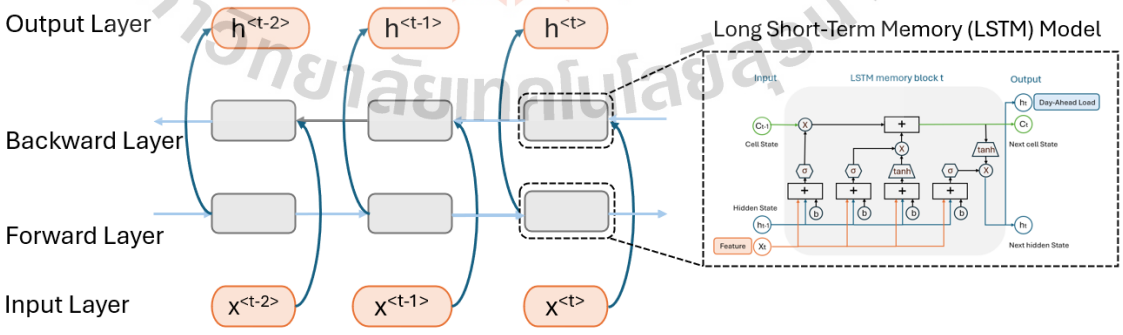


Figure 3.2 Bi-LSTM architecture

3.5.3 GRU Architecture

GRU resembles LSTM, which regulates data flow using gates. But with a simpler design, GRU provides an improvement over LSTM. GRU just has a hidden shape, not a distinct cell state like LSTM. The GRU has unrestricted access to all concealed states. Compared to LSTM units, GRUs are easier to train and do not require a memory unit to regulate the data flow. As illustrated in Fig. 3.3, GRU may train more quickly or employ less data to generalize since it has fewer parameters.

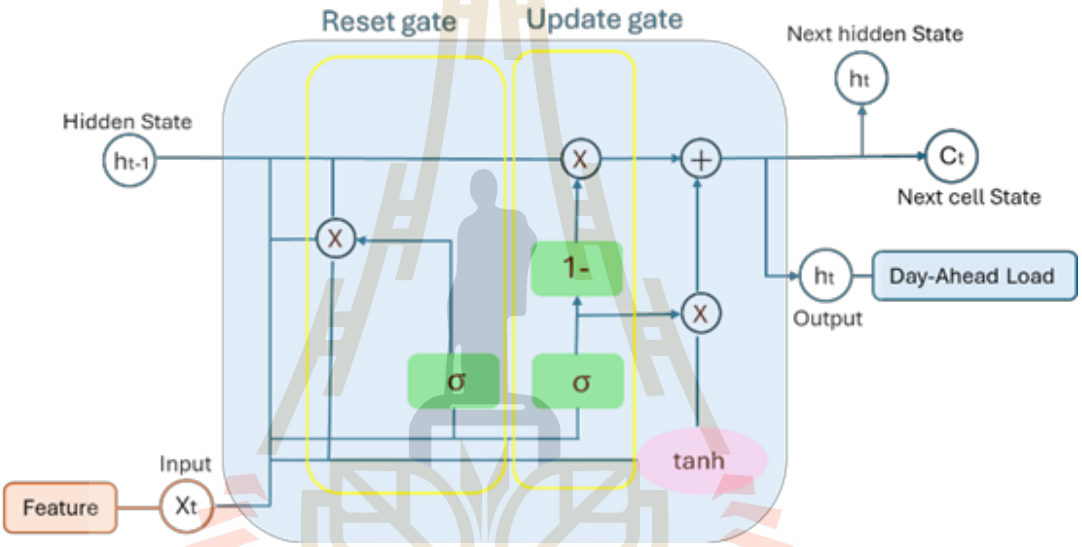


Figure 3.3 GRU architecture

3.5.4 Bi-GRU Architecture

Bidirectional GRU - A bidirectional GRU, often known as a Bi-GRU, a bidirectional GRU is designed to analyze univariate time series data. This enables the temporal dependencies found in the electrical load data to be accommodated. This integration becomes essential for precise forecasting by utilizing data from both previous and subsequent time periods. As illustrated in Fig. 3.4, this functionality plays a substantial role in improving forecasting precision.

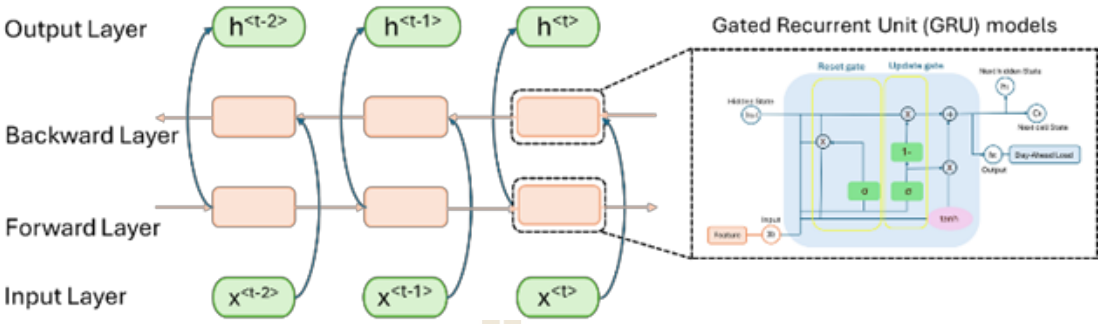


Figure 3.4 Bi-GRU architecture.

3.6 Workflow of the Proposed Method

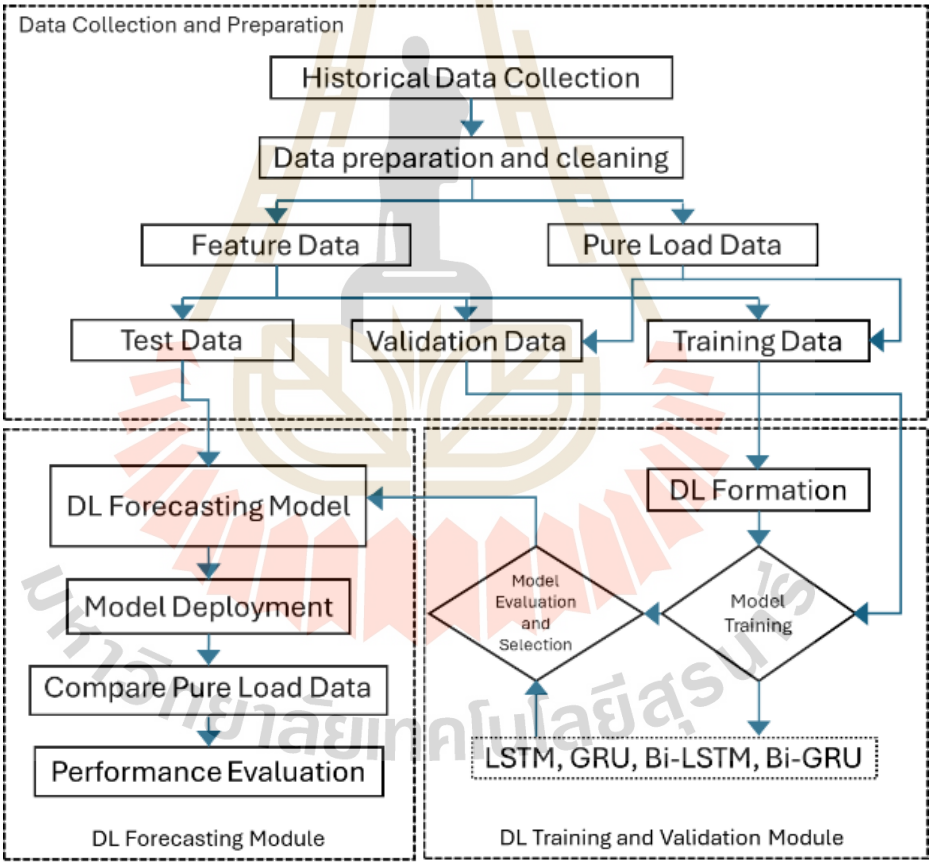


Figure 3.5 The Proposed Load Forecasting Workflow

The electrical load forecasting, including finding the model with the appropriate accuracy for the university's load data, can be done as follows. Start by collecting all relevant historical data, then clean the data with necessary techniques to facilitate

model learning. The data is divided into three groups: the training data group receives feature data as input and pure load data for the model to learn the relationship, then performs the deep learning formation step, which is to design the structure and select the function for the various models and train the model. The models used for training include LSTM, GRU, Bi-LSTM, and Bi-GRU, and they output the load forecast. The validation data group receives the same input as the first group, receiving various models for forecasting. When the output is the load forecast, it will help check the accuracy of model training and prevent overfitting in the model training process. It also helps evaluate the performance of the model during model training of the predicted values and actual load values. Then select the best-performing model as the deep learning forecasting model. The test data group receives features as input, inputs data into the selected model, and outputs daily load forecasts and obtains the developed model. By comparing the predicted values with the actual load values available, the performance of the tested model can be evaluated, as can be seen in Fig. 3.5.

3.7 Performance Factor

To examine the performance of the proposed forecasting model through evaluation metrics including percentage of MAPE, MAE and RMSE, as shown in Table 3.1. Let Y_i denote the actual load at time i, and $\hat{Y}_i$ denote the corresponding predicted load.

Table 3.1 Performance Factor

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y}_i)}{Y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert}{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.8 Simulation Results

This section presents experimental results and performance analysis of the electrical load forecasting models developed in Chapter 3. It compares the forecasting results of LSTM, GRU, Bi-LSTM, and Bi-GRU models on pre-prepared test datasets, using RMSE, MAE, and MAPE performance indicators to measure model accuracy across multiple dimensions. The analysis is divided into three main parts: evaluating model learning and accuracy during the validation and training phases and comparing forecasting results between weekdays and weekends. This chapter aims to answer the key question of which model is best suited for electrical load forecasting in the context of real-world data in terms of accuracy, consistency, and operational capability under various scenarios, and to evaluate the performance of the best model on different datasets in terms of time and context.

3.8.1 Experimental setup

Python 3 software was employed to simulate the Suranaree University of Technology's daily electrical load forecast in order to create a model suitable for the university's electrical loads and evaluate and contrast efficiency using evaluation metrics including MAPE, MAE, and RMSE. The data selected for use was every 15 minutes throughout 2023 and contained, in addition to pure load and feature or electric load data used for forecasting, The models made use of to simulate the results were LSTM, GRU, Bi-LSTM, and Bi-GRU. To get this data ready for entry into the model, it was separated into three sections. This section will show the outcomes of the simulation and different model settings.

In preparing data for power load forecasting, this research addresses missing data using filler methods that correspond to the temporal structure of the data. However, data filler can affect model performance, potentially reducing power load fluctuations and causing bias in the learning process, especially when the filled data is large or concentrated over the same time period. Furthermore, filler values can impact the generation of temporal features, such as lag features, which rely on data continuity. To limit these effects, this research controls the proportion of filled data to

a low level and selects filler methods appropriate to the data characteristics. However, future research should develop more advanced methods for handling missing data or perform model sensitivity analysis to filler to improve forecasting reliability.

The selection of hyperparameters was carried out using a trial-and-error method within a systematic evaluation framework. It began by referencing commonly used values from relevant research, then adjusting parameters individually or in sets to study their impact on the model's learning ability and forecasting errors. This method allowed for observation of the model's behavior under different settings and clearly identified trends in values affecting model performance. The final hyperparameter selected is the set of values that yields the best accuracy results, consisting of the following: A small data set of size four is used to improve the diversity of learning, using 50 epochs to give the model enough time to learn from the training data, and the model structure is set to have 100 units as the number of units in the hidden layer to train different DL models. Two layers of each model are used, a dropout rate of 0.2 is used to prevent overfitting, and Adam Optimizer is used to adjust the model weights, and a learning rate of 0.001 is used as the learning rate for Adam Optimizer, as shown in Table 3.2.

Table 3.2 Model parameter settings

Methods	LSTM	GRU	Bi-LSTM	Bi-GRU
Layer	2	2	2	2
Units	100	100	100	100
Dropout rate	0.2	0.2	0.2	0.2
Learning rate	0.001	0.001	0.001	0.001
Batch Size	4	4	4	4
Epochs	50	50	50	50
Optimizer	Adam	Adam	Adam	Adam

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.8.2 Model performance in validation and model training

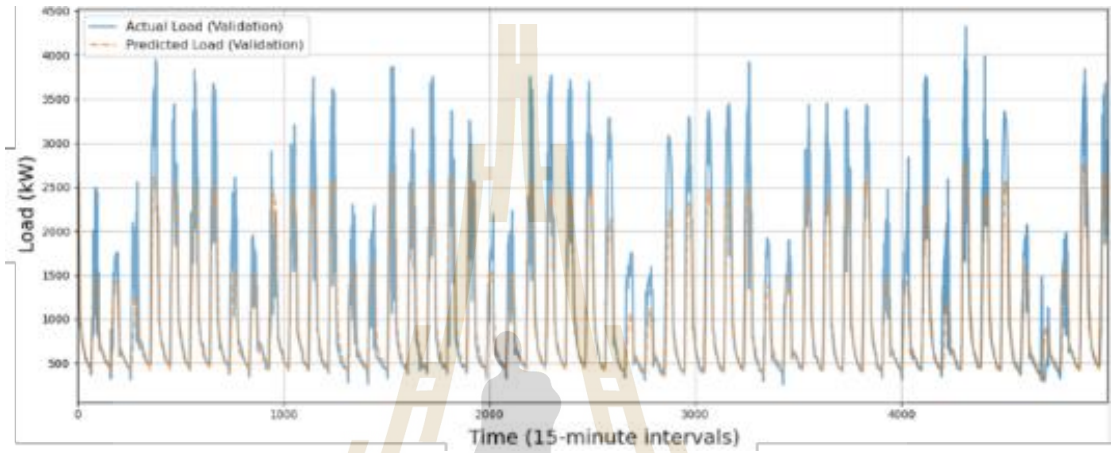
To evaluate the performance of the developed daily electrical load forecasting model in this research, the RNN family models, namely LSTM, GRU, Bi-LSTM, and Bi-GRU, were tested on the real load dataset to compare the model accuracy in the validation phase. In addition, the model training time was recorded to compare the time efficiency between different architectures. The results are shown in Table 3.3, reflecting the forecasting capability of each model under the same context and supporting the selection of the most suitable model in terms of accuracy and performance.

Table 3.3 Evaluation results of model performance in validation metrics and model training time.

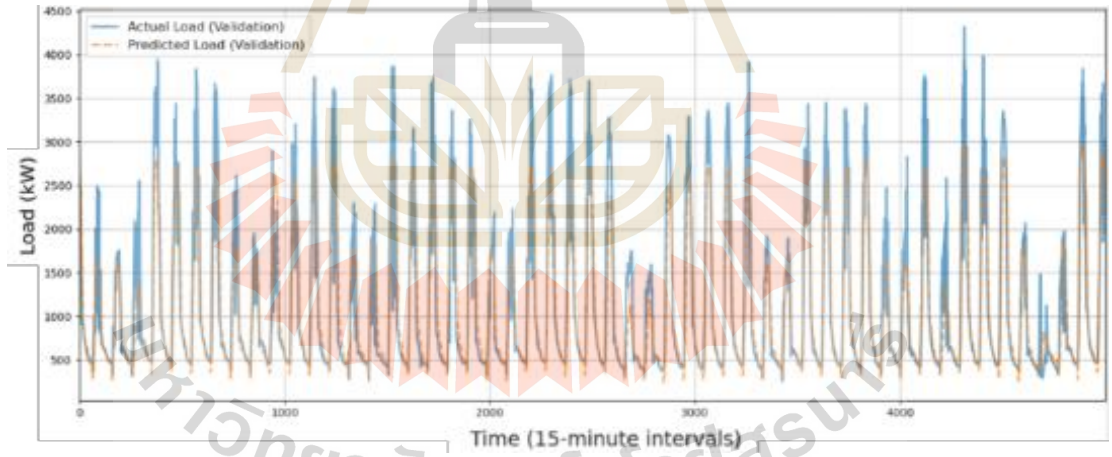
Methods	Validation Metrics			Model Training
	MAPE (%)	MAE (kW)	RMSE (kW)	Time (s)
LSTM	16.36	227.48	428.84	296.34
GRU	20.61	271.96	464.28	176.76
Bi-LSTM	16.86	241.22	439.58	589.14
Bi-GRU	17.71	252.18	453.94	318.95

The GRU model trained faster than the LSTM model, which took 296.34 seconds. Along with the evaluation of the model's performance in the validation set data, the model training time could indicate the device's usage of resources. The findings of the model's electrical load forecasts will be validated using this data. With MAPE = 16.36%, MAE = 227.49 kW, and RMSE = 428.85 kW, the LSTM model performs best after the performance evaluation. The Bi-LSTM model, with MAPE = 16.86%, MAE = 241.22 kW, and RMSE = 439.5889 kW, is the experimentally efficient model. The two models' training errors are comparable to those of the GRU model (MAPE = 20.61%, MAE = 271.96 kW, and RMSE = 464.29 kW) and Bi-GRU (MAPE = 17.71%, MAE = 252.19 kW, and RMSE = 453.94 kW), according to the three metrics. According to the validation dataset, results demonstrate the extent to which the LSTM and Bi-LSTM models learn

and capture relationships. With its speed and predicting accuracy, LSTM offers potential for real-world use. Table 3.3 displays the values in the validation set, which includes the model training settings or the simulation results of the validation metrics' performance.



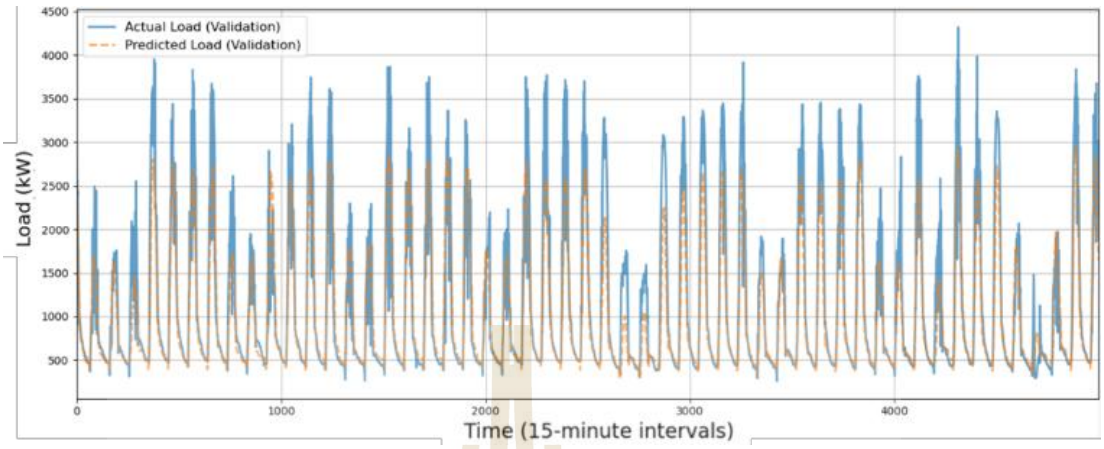
(a)



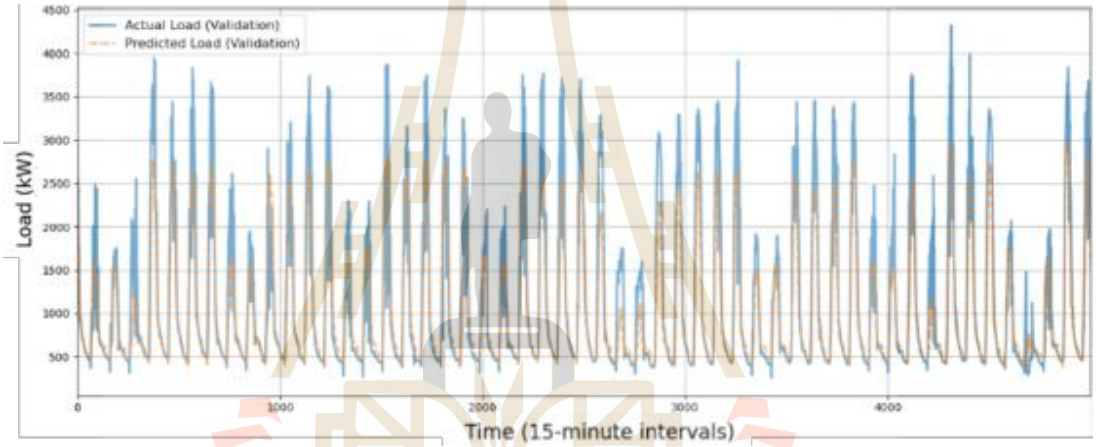
(b)

Figure 3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models

มหาวิทยาลัยเทคโนโลยีสุรนารี



(c)



(d)

Figure 3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models

Fig. 3.6 presents the prediction results between the actual load and the model's predicted load on the validation data. It includes the predictions of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. The results of the prediction align with the mentioned earlier metrics.

3.8.3 Performance of each model on weekday and weekend loads

After model training and validation, the model is tested in real use. The test set data is divided into two types: weekday load is the load from Monday to Friday, which has a higher total electricity demand than the holiday load. Additionally,

weekend load is the load on Saturday or Sunday with the largest variability in electricity demand. The following test days have been selected to evaluate the precision and capacity for forecasting each load model when the electrical load exhibits distinct characteristics: The weekday load is Wednesday, March 1, 2023. And the weekend load is Sunday, October 1, 2023. Forecasting the electrical load output for a day or 24 hours in the future is required, with data being displayed every 15 minutes. There will be a total of 96 data sets. The actual test date is simulated through each model. The results obtained from simulating the weekday load are shown on Fig. 3.7 and the results from simulating the holiday load are shown on Fig. 3.8. The forecast graph will appear.

The blue line represents the actual electrical load, or pure load, the yellow line, the green line, the red line, and the purple line. Shows the forecast of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. In the weekdays load forecast, the LSTM model graph gives the forecast that is closest to the actual load graph. The graph is closely related to the actual load graph. The next order has the forecast that deviates from the actual load, which are the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively. However, other models can provide the forecast of the actual load trend. In the part of the weekend's load forecast, the LSTM model graph still gives the forecast that is closest to the actual load graph. However, there are still some load periods that do not give the forecast that is as close to the actual value as it should be because it is a day with a significant load fluctuation. Overall, the LSTM model still performs well. The models whose forecast graphs are closest to the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively.

มหาวิทยาลัยเทคโนโลยีสุรนารี

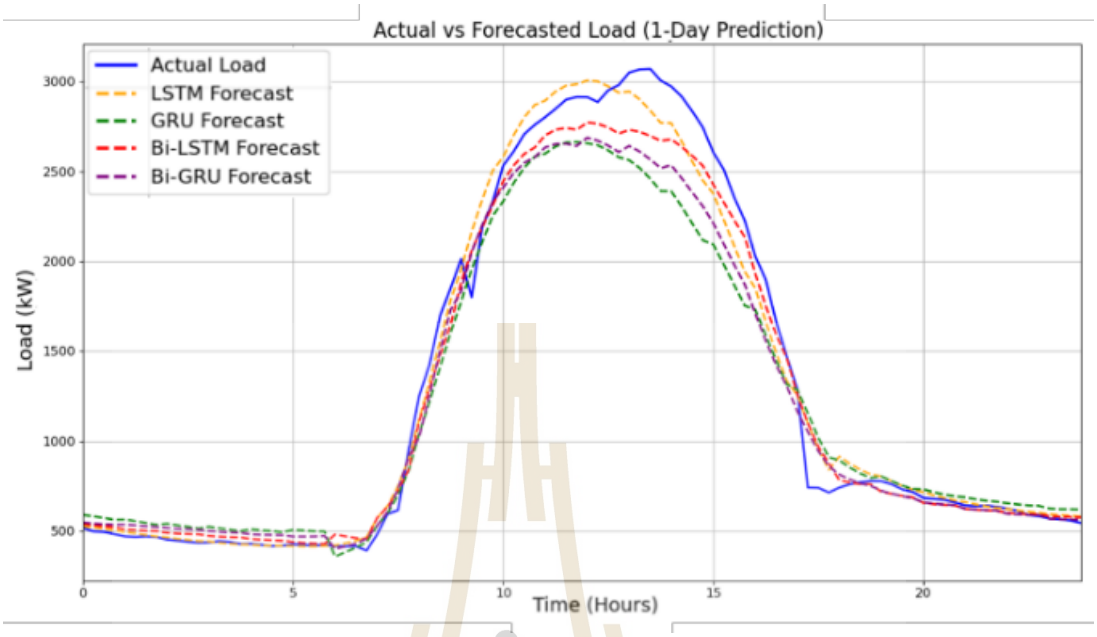


Figure 3.7 One-day-ahead electricity load forecast for each model on weekdays

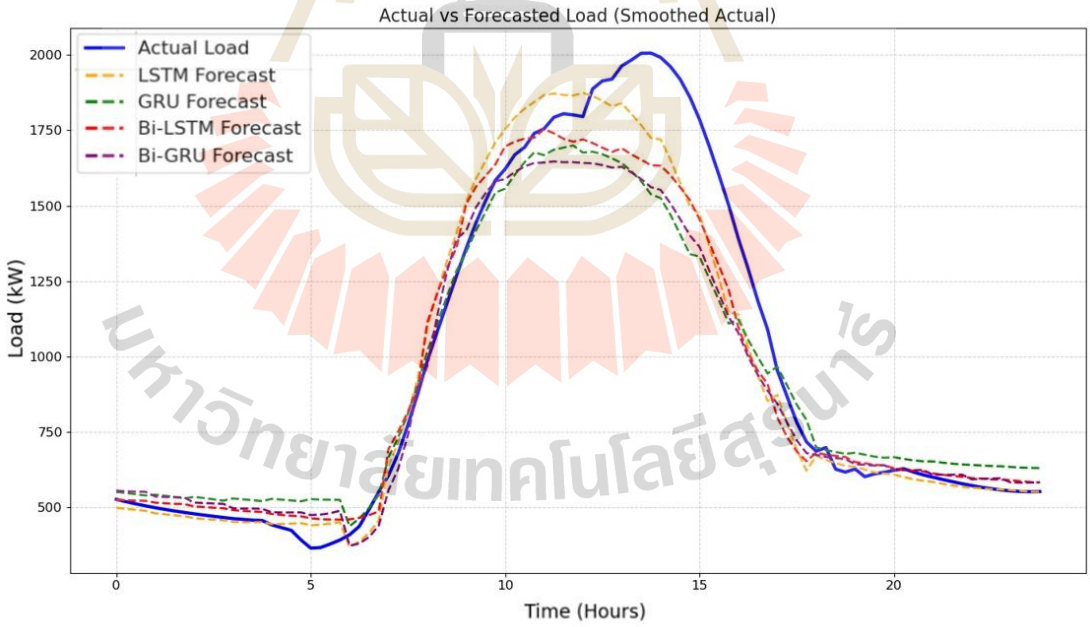


Figure 3.8 One-day-ahead electricity load forecast for each model on weekends

Table 3.4 Performance of each model on weekday load

Methods	Model Performance of Weekday Load		
	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	**6.09**	**81.51**	**122.84**
GRU	13.29	170.33	242.46
Bi-LSTM	7.38	91.38	131.30
Bi-GRU	9.88	135.16	196.28

Table 3.5 Performance of each model on weekend load

Methods	Model Performance of Weekend Load		
	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	9.62	115.12	214.53
GRU	15.51	157.18	254.17
Bi-LSTM	11.21	127.00	219.68
Bi-GRU	11.42	138.39	243.42

In addition, the performance of each model was evaluated through all three indicators or through evaluation metrics. The results of the weekday load evaluation are as follows: The LSTM model has a MAPE = 6.09%, MAE = 81.51 kW, and RMSE = 122.84 kW. The GRU model has MAPE = 13.29%, MAE = 170.33 kW, and RMSE = 242.46 kW. The Bi-LSTM model has a MAPE = 7.38%, MAE = 91.38 kW, and RMSE = 131.30 kW. And the Bi-GRU model has a MAPE = 9.88%, MAE = 135.16 kW, and RMSE = 196.28 kW. Shown in Table 3.4. The results of the holiday load evaluation are as follows: The LSTM model has a MAPE = 9.62%, MAE = 115.12 kW, and RMSE = 214.53 kW; the GRU model has a MAPE = 15.51%, MAE = 157.18 kW, and RMSE = 254.17 kW; the Bi-LSTM model has a MAPE = 11.21%, MAE = 127.00 kW, and RMSE = 219.68 kW; and the Bi-GRU model has a MAPE = 11.42%, MAE = 138.39 kW, and RMSE = 243.42 kW. Shown in Table 3.5. From the evaluation of the model performance, the results of the weekday load, the LSTM model shows the lowest forecast error when compared to the Bi-LSTM model, Bi-GRU model, and GRU model. For the results of holiday load,

the LSTM model still has the lowest error of all metrics when compared to other models.

However, it still has a higher error than the weekday load because it is a day with high volatility, especially in hours 13 to 18, which has the highest volatility, resulting in the highest forecast error during that period. Overall, the LSTM model still has the best power demand prediction and gives the lowest error for that day. The results show that the most efficient and accurate model for university load data is the LSTM model because the model can learn the patterns and relationships of long-term periods well, including managing the time series data in the data set most efficiently. The weekday load pattern has slightly higher forecast accuracy than the holiday load because the holiday load is more volatile. The next most efficient models are the Bi-LSTM and Bi-GRU models because the type of data that needs to be forecasted may not give enough weight to consider the data from the past and the future, causing the model to learn the relationship and not be able to use its full potential. The last model is the GRU model because it is a simpler model than the LSTM model, which makes it not as good at learning the relationship of the data set, and this type of data may not be suitable for the model. While Suranaree University's power load can be reliably predicted by LSTM and Bi-LSTM models, there are still some restrictions that should be considered when using these models in other situations.

To properly understand long-term correlation patterns, the models need a lot of high-quality data. In the case of data-constrained environments, such as small-scale power systems in rural areas or organizations with data shortages, the model performance may be reduced. In addition, Bi-LSTM and Bi-GRU models are complex and require higher computing resources than GRUs, which may not be suitable in systems with limited computing resources, such as IoT devices or edge computing systems. However, the models can be applied in different environments, both in the large-scale energy industry and in distributed energy systems, by adjusting the model structure and parameter settings to suit the data complexity in each context or adding variables such as weather data or sunrise-sunset times to increase the accuracy of predicting fluctuating energy usage. The use of GRU models with higher model training speed in real-time systems may be a suitable option. The results of this

study have important implications for the energy sector in many dimensions. The developed electricity load forecasting model can help enhance the efficiency of energy management strategies in various contexts. By precisely predicting energy demand ahead of time, energy suppliers may effectively plan power generation and distribution, minimize overloading, and minimize transmission line losses through integration with smart grid energy management technologies. These models can be used in industrial energy management to predict the energy demand for pieces of equipment or production processes. This helps energy managers plan when to use energy (demand-side management), lower electricity costs during periods of high energy prices, and make the production system more sustainable. In addition, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources such as solar and wind power, which can help reduce energy imbalances and improve the efficiency of renewable energy use, as well as support the reduction of unnecessary energy use as part of the strategy to reduce greenhouse gas emissions in the energy sector.

3.8.4 Performance of LSTM model on 12 test sets

In power forecasting for systems with continuous load variability, the use of machine learning models, especially Long Short-Term Memory (LSTM) models, is a popular approach that shows good performance in many contexts. This is because LSTMs have the ability to remember data sequences and detect long-term patterns in data. Testing the LSTM model on datasets with different characteristics in terms of energy consumption behavior, such as holidays, exam days, or different seasons, can help to clearly assess the model's flexibility and accuracy. The datasets were managed by performing feature engineering before importing the model. Therefore, the analysis in this section focuses on comparing the performance of the LSTM model on 12 test datasets, which reflect different temporal characteristics, along with the use of performance indicators such as MAPE, MAE, and RMSE. The performance evaluation results of the LSTM model on 12 test datasets, each of which represents a variety of time periods or situations, such as long holidays, school openings, exam days, or

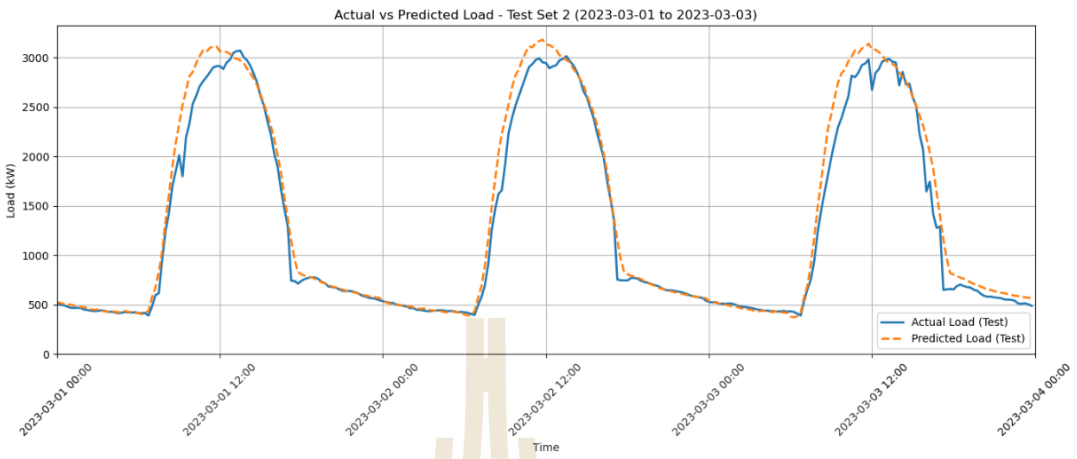


Figure 3.9 Forecasting Result – Test Set 3: Exam Period

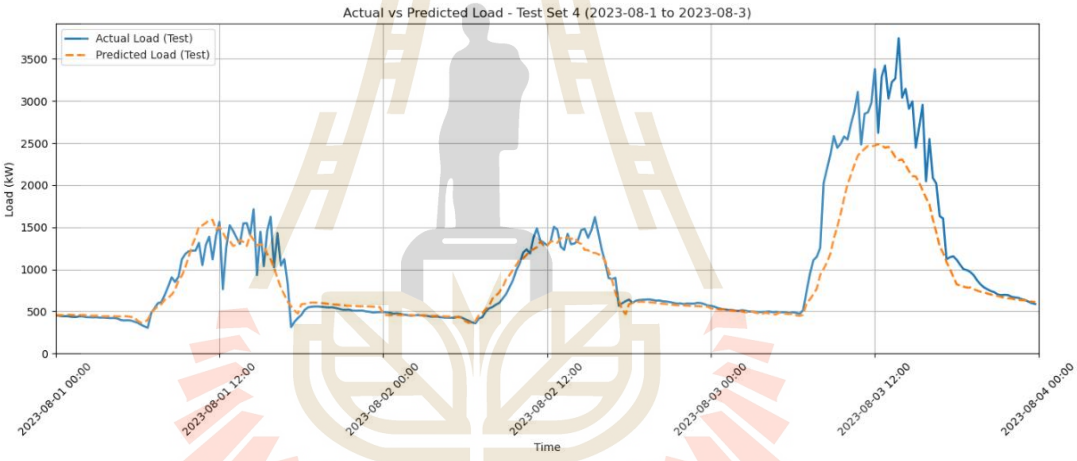


Figure 3.10 Forecasting Result – Test Set 6: Start of the Semester

different seasons, are presented. Each data set consists of 3 days of data to clearly show the trends in energy consumption behavior in each period, The results are shown in Appendix A.

From the evaluation results, it was found that Set 3 (exam period 2023-03-02) gave the lowest MAPE value at 5.93%, indicating the highest forecast accuracy with MAE equal to 83.05 kW and RMSE equal to 145.60 kW, which showed good results in terms of percentage of error. Meanwhile, set 6 (the start of the semester, 2023-08-02) gave the lowest MAE and RMSE values, namely MAE = 59.18 kW and RMSE = 93.66 kW, indicating that in terms of absolute error and root mean square error, the model

can predict the closest to the actual value. For analysis, the forecast results of Set 3 and Set 6 are presented in graph form. The graph shows the difference between the actual value and the value predicted by the model, helping to show the level of model consistency in each period shown in Fig. 3.9 and Fig. 3.10.

Table 3.6 Model Performance on Validation and Average Performance on 12 Test Sets.

Methods	Model performance of the validation set			Average performance of the model on 12 test sets		
	MAPE (%)	MAE (kW)	RMSE (kW)	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	13.71	177.58	315.67	14.07	188.77	302.04

In addition to the evaluation of each dataset, Table 3.6 shows the overall evaluation results in two aspects: the prediction results on the validation set and the average model results from testing on the 12 datasets. On the validation set, the LSTM model yielded MAPE = 13.71%, MAE = 177.58 kW, and RMSE = 315.67 kW. The average of the model on the 12 test data sets yielded MAPE = 14.07%, MAE = 188.77 kW, and RMSE = 302.04 kW. These results reflect that the model can maintain a stable level of accuracy when comparing the validation set and the test set, with an acceptable level of error for use in short-term forecasting systems. From the evaluation of the LSTM model's performance on various data sets in terms of time and context, it was found that the model can forecast accurately in some time periods, especially during exams and school openings, where the forecasting results are more accurate than other periods. The use of lag features and clustering data by time characteristics enhances the model's potential to capture the patterns of load data. In addition, the overall evaluation of the validation set and the average values from the test sets confirm that the model is stable and can be effectively applied to load forecasting in various contexts.

3.9 Summary of the daily forecast

This article presents an alternative approach to developing short-term electric load forecasting models and improving their accuracy through deep learning techniques, including the Day-Aware Data Arrangement method. The most widely used forecasting methods in this field were simulated and compared to evaluate their performance and accuracy. The daily electric load forecasting for Suranaree University of Technology often correlates with past time intervals, allowing LSTM to effectively capture these patterns better than other models. As a result, its load predictions are the closest to the actual values. Adding bidirectional layers to the models did not significantly enhance the results for electric load data. The GRU model, while easier to implement and faster to process, demonstrated lower performance and accuracy compared to the LSTM model for sequential data of this nature. Notwithstanding the encouraging outcomes, this study also discusses significant shortcomings and challenges for further applicability. For maximum performance, high-quality and adequate data are necessary, which might be difficult in contexts with limited resources. Nonetheless, similar models can be used in various energy sectors by reorganizing them, adding new variables, and adjusting the parameter values. The study's findings have important ramifications for the energy industry overall. Planning energy production and distribution effectively, reducing energy losses, and maximizing the integration of renewable energy sources are all made possible by the inclusion of accurate forecasting models, particularly LSTMs, into energy management techniques. Faster GRU models may be suitable for real-time applications where computational resources are limited. And adding Feature Engineering to evaluate the stability of the LSTM model on 12 test sets to study the variability of the results that may be caused by different data characteristics at different times.

CHAPTER IV

Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus

4.1 Introduction

This section proposes an approach to developing a one-day-ahead electrical load forecasting model using deep learning techniques. It compares deep learning models with basic models, namely Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM, to improve accuracy and reliability through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis. This approach utilizes data collection and feature engineering to optimize the context of an electrical load with integrated solar power generation, Using Suranaree University of Technology (SUT) campus electrical system as a case study. Three accuracy indices are examined: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). Among the NN-based models, Bi-LSTM and LSTM tend to provide the best overall forecasting performance from the input data, and the Bi-LSTM model is the most accurate model with the lowest metric value among all models. The lag load feature, or the value of the electrical load in the past one day, is the feature that is most related to the forecasting target and has the most impact on the model's decision. Parts of this thesis have been published in ECTI Transactions on Electrical Engineering, Electronics, and Communications (2025), is shown in the Appendix D, and another set of simulated data is presented to further enhance the reliability of the framework, also shown in this section.

4.2 Data Collection and Preparation

Before entering the model for load forecasting, Suranaree University of Technology's electricity consumption data was collected for 1 year, collected every

15 minutes in 2023, totaling 35,040 data sets, which is considered high-resolution data for load forecasting. The data was collected from Automatic Metering (AMR) maintained by the Provincial Electricity Authority (PEA). In addition, solar energy generation data was collected from an Automatic Metering (AMR) maintained by an external company, ENOS, with the same resolution as the electricity load data. Weather data and academic calendar data for that year were collected and prepared to input all this data into the future feature engineering method. The required data is the actual load data. Therefore, the load obtained from the meter was combined with the solar energy generation data to obtain data showing the actual load of the university.

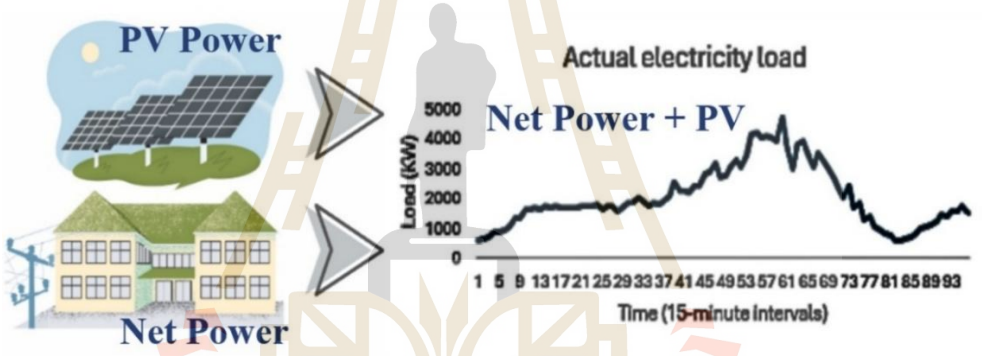


Figure 4.1 Preparation of Actual Electricity Load Data

The preparation of the real electricity load data can be shown on Fig. 4.1. All data used as features and output for the model to learn were divided into two parts: the first part was the data set to be trained, and the other part was partitioned for model validation. Time Series Cross-Validation (TSCV) was performed on this data set, which is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. The training and validation sets were generated using a Time Series Split with 5 folds. The second part was the data set tested on the model after training and validation were completed. It was a data set that the model had never seen before, so the results were unbiased and divided into 4608 data points. For testing, this group of data will be divided into 10 sets; each set

will have 3 days. Each set is divided according to the types of data groups, namely long holiday, weekdays during the school term, exam period, summer season, the start of the semester, compensation holiday period, rainy season, weekdays and weekends, term break, and winter season.

4.3 Feature Modeling and Representation of Electrical Load Patterns

Feature engineering has a group of data that has the characteristics of feature creation, which is creating new features from existing time data and converting time data that may not be easy for the model to logical data with contextual meaning.

4.3.1 SUT's Campus Data

It consists of the following techniques: In the feature transformation method, features that use this method include Weekdays and Weekend, Daytime and Nighttime, Compensatory Holiday, and Long Weekend, which are converting to variable values and binary representation with numbers 0 and 1. In the time-based feature engineering method, features that use this method include yearly hours, which divides the data into meaningful time sequences with 1 year's data every 15 minutes, creating features that allow the model to learn the day and time. The Time Component Extraction method features that use this method include Quarter-Hourly Interval, which extracts time components into separate features, creating new features that describe the sub-patterns that occur in each hour and the lag features method. Features that use this method include Lag Load, which creates features from previous values by taking historical load values to create features. In addition, there is a group of characteristics: external features. External variables affect the forecast of the target variable, such as ambient temperature, an external factor that directly affects energy use behavior, and dew point, an indicator of air humidity.

4.3.2 New England (ISO-NE) Data

Time-based feature generation plays a crucial role in enabling the model to efficiently learn periodic patterns and long-term trends of electrical load in the second dataset. Additional relevant features can be generated as follows:

The "Day" feature is used to reflect the order of days within a month, allowing the model to recognize changes in electricity usage behavior that may occur at the beginning, middle, or end of the month, such as during payday or before long holidays. The "Month" feature represents the seasonality of electrical load, particularly the impact of different weather conditions throughout the year, such as summer or rainy season, which directly influences electricity demand. The "Year" feature plays a role in reflecting long-term trends in electrical load, such as increases in the number of electrical appliances, infrastructure expansion, or changes in energy policy. This allows the model to appropriately differentiate electrical load behavior over several years. Furthermore, the "Weekday" feature is used to distinguish between weekdays and holidays, a key factor clearly influencing electricity usage patterns. Generally, electrical load patterns on weekdays differ from holidays, both in terms of load levels and peak times. Incorporating this feature allows the model to better learn about the differences in electricity usage behavior across different types of days.

In summary, adding time-related features—Day, Month, Year, and Weekday—along with Lag Load, Ambient Temperature, and Yearly Hours features in the ISO-NE dataset for further simulation enhances the model's ability to capture temporal relationships, long-term trends, and periodic patterns of electricity load—critical components for accurate and stable electricity load forecasting in the context of time series data.

4.4 Feature Selection and Mutual Information

Mutual Information (MI) captures the relationship of input features to a non-linear data model, shows their importance, and indicates how the feature reduces uncertainty with respect to the forecasting target. It is used as a tool for selecting appropriate forecasting features, especially in tasks where the number of features varies without necessarily being linear. It is calculated using the following equations (4.1).

$$i(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \cdot log\left(\frac{p(x,y)}{p(x) \cdot p(y)}\right) \quad (4.1)$$

where Y is the target variable, i.e., the future load, X is the input variable, $p(x, y)$ is the joint probability of x and y, and $p(x)$ and $p(y)$ are the marginal probabilities of each variable. The value of MI is expressed as a real number greater than or equal to 0. The higher the value of MI, the more the feature is related to the forecast value. MI can help reduce the size of the data to control both the speed of training and the interpretation of the model. MI is a powerful tool for feature selection for time series models, especially in complex load forecasting tasks that change over time and in different contexts. Selecting appropriate features using MI can make the model more accurate and easier to interpret.

Feature selection in load forecasting each feature plays a physical or behavioral role in load patterns. The Lag Load feature was selected because electrical loads are similar and have daily activities that often repeat within 24 hours, making it a very important representative of daily patterns in forecasting. The Yearly Hours and Quarter-Hourly Interval features help the model understand the time sequence of each day and allow the model to recognize period differences. When used in combination with other features, the model can better learn about daily trends. The Weekdays and Weekend features were selected because electricity usage by electricity consumers differs significantly during holidays and weekdays. The Daytime and Nighttime features were selected because people's behaviour shows higher electricity usage in the morning than at night, so the model should be able to see this difference. The Compensatory Holiday and Long Weekend features have different load patterns during long holidays and special holidays. The Ambient Temperature and Dew Point features are weather-related features that reflect the characteristics of electricity consumers when temperatures and humidity change. They also provide the model with sufficient forecasting capabilities when there is load variation caused by changing weather conditions.

4.5 Modeling Approaches and Baseline Methods

Accurate short-term electricity load forecasting requires models that can effectively capture temporal dependencies, nonlinear relationships, and variations in

consumption patterns under different operating conditions. To comprehensively evaluate forecasting performance and demonstrate the effectiveness of advanced deep learning techniques, this study investigates both baseline forecasting methods and a range of time-sequence deep learning models.

The models used to simulate results in the Recurrent Neural Network (RNN) category were compared and selected. The models with the highest potential for accuracy were LSTM and Bi-LSTM, as discussed in the previous chapter. The scale was then expanded to compare with other categories to cover a wider range of subjects. In the classical statistical time-series method category, the moving average method was selected. This is a traditional statistical forecasting method that calculates the average of the electrical load over a given period and uses that value as a forecast for the next period. It uses the average of the past 3 days. This method does not involve model learning and is used as a basic method. In the Feedforward Neural Network (FNN) category, the DNN model was selected. This is a multi-layered hidden neural network where data flows forward in only one direction from input to output, without memory of the past like RNNs. In the hybrid model category, the models selected were Attention-LSTM and CNN-LSTM.

4.5.1 Attention-LSTM Model

Attention-LSTM consists of an LSTM model that extracts deep features from time-dependent data and an attention component that determines which data in the sequence has the greatest impact on load forecasting. Features fed as input are given into the LSTM model, with the input data at each time point represented by variables X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1}, and X_t. The model then generates a hidden state (h_t) that stores in-depth information about past electrical load behavior, as shown by the equation (4.2).

$$h_t = LSTM(X_t, h_{t-1}, C_{t-1}) \quad (4.2)$$

Each timestep yields the results h_{t-4}, h_{t-3}, h_{t-2}, h_{t-1}, ht, where C_{t-1} is the cell state capable of retaining information for a long period. Since the LSTM model focuses only on the last ht, an attention mechanism is added to emphasize (weight) each hidden

state. It creates weights (α_1, α_2, α_3, α_4, α_5) indicating the importance of the information in that timestep and its impact on the forecast for each hidden state. A softmax function is then used to normalize the result so that the sum equals 1. This is shown in the equation (4.3).

$$\alpha_i = \frac{exp(score(hi))}{\sum_{j=1}^{T} exp(score(hi))} \tag{4.3}$$

Then, the hidden states are multiplied by the weight (α) and combined to create a Context Vector (C), which is the central value summarizing the historical load data that the model prioritizes most. This represents sequence data all the time and can be expressed as the follows equations (4.4).

$$C = \sum_{i=1}^{T} \alpha_i h_i \tag{4.4}$$

Once C is obtained, it is combined with the LSTM Output to create the Feature Final (Fea_{final}), which represents the features processed by both LSTM and Attention. It is then passed through the Dense Layer to predict the electrical load for the next day. The attention mechanism was implemented using Luong dot-product attention, where the attention weights (α) are normalized using the softmax function. Feafinal represents the combined feature vector generated from the LSTM hidden states and attention mechanism. The structural diagram of the model is shown in Fig. 4.2.

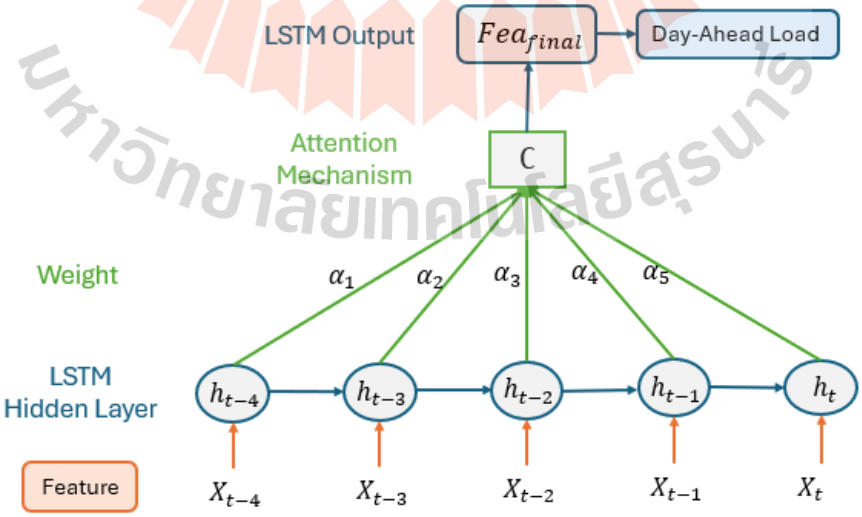


Figure 4.2 Attention-LSTM architecture

4.5.2 CNN-LSTM Model

The CNN-LSTM model is a combination of Convolutional Neural Network (CNN) and LSTM models for improved feature extraction and learning of temporal dependencies. CNN extracts feature from sliding window load data using 1D convolution to compute the feature map at position t. A kernel of size k is used, employing multiple filters to detect different patterns, as shown in the equation (4.5).

$$z_t = \sum_{i=0}^{k-1} w_i \cdot x_{t+i} + b \tag{4.5}$$

Where w_i is the filter weight (kernel weight), b is the bias, i is the index of the position within the kernel, x_{t+i} is the model input at time $t+i$, and z_t is the feature value obtained after convolution. A pooling layer is used to reduce the dimensionality of the data and extract only the important features, where p is the size of the pooling window, as shown in the equation (4.6).

$$p_t = max(z_{t:t+p}) \tag{4.6}$$

Where p_t is the pooling result at time position t, $z_{t:t+p}$ is the sequence of feature map values obtained from the convolution step of the CNN from time t to $t+p$, and $max(\cdot)$ is the max pooling function. After the CNN model finishes extracting features, it sends the obtained features to the LSTM model to learn the time sequence and finally sends them to the Dense Layer to forecast the electricity load for the next day. The predicted load for the next day is given by the equation (4.7).

$$\widehat{Y_{t+1}} = W_d \cdot H_t + b_d \tag{4.7}$$

This is the step for creating the output layer of the model, where $\widehat{Y_{t+1}}$ is the value, the model predicts at the next time point $t+1$, and H_t is the hidden state of the LSTM at time t, which is the result of the time sequence learning process. This data has first featured using a CNN for short-term spatial or pattern extraction before being fed into LSTM to learn temporal relationships. It represents temporal memory that includes past load patterns and the influence of other features. W_d is the weight matrix of the Dense layer (fully connected layer), which transforms the hidden state of the LSTM into a dimension suitable for the desired output prediction, and b_d is the bias of the output layer, which adjusts the prediction value independently of the input,

allowing the model to be more flexible in estimating the true value. The CNN-LSTM model uses CNN to extract key patterns from historical data, then uses an LSTM to learn temporal relationships, and finally uses the Dense layer to transform the hidden state into the predicted electrical load value.

4.5.3 DNN Model

As part of the Artificial Neural Network (ANN) method, it learns deeper input-output relationships, especially in tasks with nonlinear data. The model consists of three parts: the Input Layer, which receives feature data and prepares it for encoding before sending it to the next layer; next are the Hidden Layers, which learn increasingly complex data and can be divided into multiple layers to accommodate more complex incoming data; and finally, the Output Layer, which predicts the continuous load (h^l). The calculation is shown in the equation (4.8).

$$h^l = f(w^l \cdot h^{l-1} + b^l) \tag{4.8}$$

Where w^l is the weight of the l layer, b^l is the bias constant, and $f(\cdot)$ is the activation function.

4.5.4 Moving average

This is a traditional, basic method used to forecast electrical loads with relatively stable patterns and minimal data fluctuations. The type used is the Simple Moving Average (SMA). Future forecasts depend on the average load usage over similar past periods, focusing on patterns and trends. The forecast value for the current day at time t will be equal to the average of the past three days at time t, as shown by the equation (4.9).

$$\widehat{Y_{d,t}} = \frac{Y_{d-1,t} + Y_{d-2,t} + Y_{d-3,t}}{3} \tag{4.9}$$

Where $\widehat{Y_{d,t}}$ is the forecast value for day d at time t and $Y_{d-1,t}$ is the actual load value for the previous days at time t.

4.6 Time Series Cross-Validation (TSCV)

Model performance evaluation is an important part of the data analysis process, especially in tasks involving time series data such as load forecasting, which are characterized by temporal order and temporal dependency, making it impossible to use traditional cross-validation techniques that are applied to independent and identically distributed data. To solve this problem, Time Series Cross-Validation (TSCV) is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. In conventional cross-validation, such as K-Fold Cross-Validation, data is randomly partitioned regardless of the order of the data. This will result in future data being present in the training set and historical data being present in the test set, resulting in data leakage that makes the model evaluation results do not reflect the true performance. TSCV solves this problem by preserving the order of the data and using a "walk forward" or "rolling window" approach instead. Here we will discuss the expanding window (walk-forward validation) approach, which starts with a small training data set and gradually adds new data to the training set, while the test set always moves in time. This is suitable for models that need to "learn" from the past and predict the future, especially in contexts where the data tends to change cumulatively, as shown in Fig. 4.3.

Fold	Train Index	Validation Index
1	[0, 1]	[2, 3]
2	[0, 1, 2, 3]	[4, 5]
3	[0, 1, 2, 3, 4, 5]	[6, 7]
4	[0, 1, 2, 3, 4, 5, 6, 7]	[8, 9]

Figure 4.3 Expanding Window

มหาวิทยาลัยเทคโนโลยีสุรนารี

Time Series Cross-Validation is an important process that enables accurate model evaluation in the context of temporal data, especially in load forecasting applications where data are time-varying and have a seasonal structure. Proper selection of TSCV prevents data leakage, increases the reliability of evaluation results, and supports the creation of models that can be confidently put to practical use.

4.7 Shapley Additive Explanation (SHAP)

In a load forecasting system that uses complex models, users often need not only accurate predictions but also to understand why the model produces those predictions in order to build confidence, support business decisions, and verify the stability of the model when the environment changes. SHAP is one of the tools that can provide a rational and easily interpretable explanation of these models.

SHAP is a method based on the Shapley Value from Cooperative Game Theory, which measures how each feature contributes to the model's decision. SHAP transforms the model description problem into a game in which each feature is a player, and the model's outcome is a reward that must be shared (prediction). The Shapley Value calculation considers all subsets of features (feature coalitions) to assess the impact of each feature. The calculation principle (conceptually) can be described as the Shapley value of feature i is defined by the following equation (4.10).

$$\emptyset i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)] \qquad (4.10)$$

where N is the set of all features, S is the subset of features excluding i, $f(S)$ is the predicted value of the model when using only the feature in S, and $\emptyset i$ is the average of the effects that feature i has on the predicted value when combined with all other features in various orders.

In load forecasting, the input features can be complex. SHAP can analyze how much historical load (lag features) or temperature influence each time period, interpret which factors cause the model to overproduce or underproduce the forecast, and

identify globally and locally important features, or analyze both overall and case-specific important features. SHAP is the current model explainer, especially for load forecasting tasks that require both accuracy and transparent understanding of the model. It allows us to look into the black box and distinguish how each feature affects the outcome, which is in line with the goal of explainable Artificial Intelligence.

4.8 Forecasting Process Description

The development of an accurate and interpretable short-term load forecasting model involves a series of systematic processes, each contributing to the robustness and practicality of the final solution. This research was carried out in multiple stages, starting from the preliminary design and comparison of deep learning models and progressing through advanced techniques for feature selection, model validation, and explainability. The overall workflow can be divided into three main phases, each emphasizing different aspects of model enhancement: initial development, performance refinement, and interpretability.

Phase 1: Initial Development and Model Comparison The research began with data collection and preprocessing, focusing on historical load data from Suranaree University of Technology, complemented by contextual variables such as temperature, humidity, calendar effects, and time indicators. The raw data were cleaned, normalized, and structured into a time series format suitable for model training. Subsequently, feature engineering was employed to capture the temporal characteristics of the dataset. Manually crafted features, such as day of the week, hour of the day, holiday indicators, and seasonality factors, were added to enrich the dataset. These features are known to have a significant impact on electricity consumption in institutional settings. The core of this phase was the design and training of four deep learning models. These models were selected for their proven effectiveness in modeling time-dependent patterns. After training the models on the same dataset, their performance was evaluated using standard metrics, including Root

Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The comparison results consistently show that this model outperforms other models, especially during periods of high workload volatility. Therefore, it is selected as the most suitable model for further development.

Phase 2: Feature Selection and Time-Aware Validation To improve accuracy further and address potential overfitting, the next phase introduced automated feature selection using the mutual information (MI) method. MI assesses the non-linear dependency between each input variable and the target load, helping to retain only those features that contribute meaningfully to prediction performance. This step allowed for dimensionality reduction while preserving the complexity needed to reflect real-world electricity usage patterns. After refining the feature set, the model was retrained using the selected features. In this phase, model evaluation was conducted using Time Series Cross-Validation (TSCV) with an Expanding Window strategy. Unlike traditional k-fold cross-validation, which can lead to data leakage in time series data, TSCV respects the temporal order of the data, simulating real-world forecasting conditions where only past information is available to predict the future. This enhanced validation approach provided a more realistic and robust assessment of the model's generalization ability.

Phase 3: Explainability and Model Transparency Although deep learning models can offer high accuracy, their black box nature often limits their practical utility in energy decision-making environments. To address this, the final phase introduced an explainability mechanism using SHAP. SHAP quantifies the contribution of each feature to a specific prediction (local explanation) and identifies the most influential features across the dataset (global importance). Insights from SHAP analysis were used to refine the feature set further. Features with low or inconsistent importance were reviewed, and the model was retrained with a final set of contextual and statistically relevant variables. This interpretability step not only improved the transparency of the forecasting system but also provided actionable insights for energy managers and

stakeholders, who could now better understand the reason behind the model's outputs.

The complete workflow—beginning with basic data preparation, progressing through model comparison, and culminating in interpretability—forms a robust framework for load forecasting that balances accuracy, transparency, and practical applicability. Each phase is built upon the previous one: initial model comparisons identify a strong baseline, advanced feature selection and time-aware validation enhance accuracy, and SHAP-based interpretability ensures model decisions can be trusted and understood. The overall research process is presented in Fig. 4.4, which illustrates the step-by-step process from data collection to descriptive load forecasting.

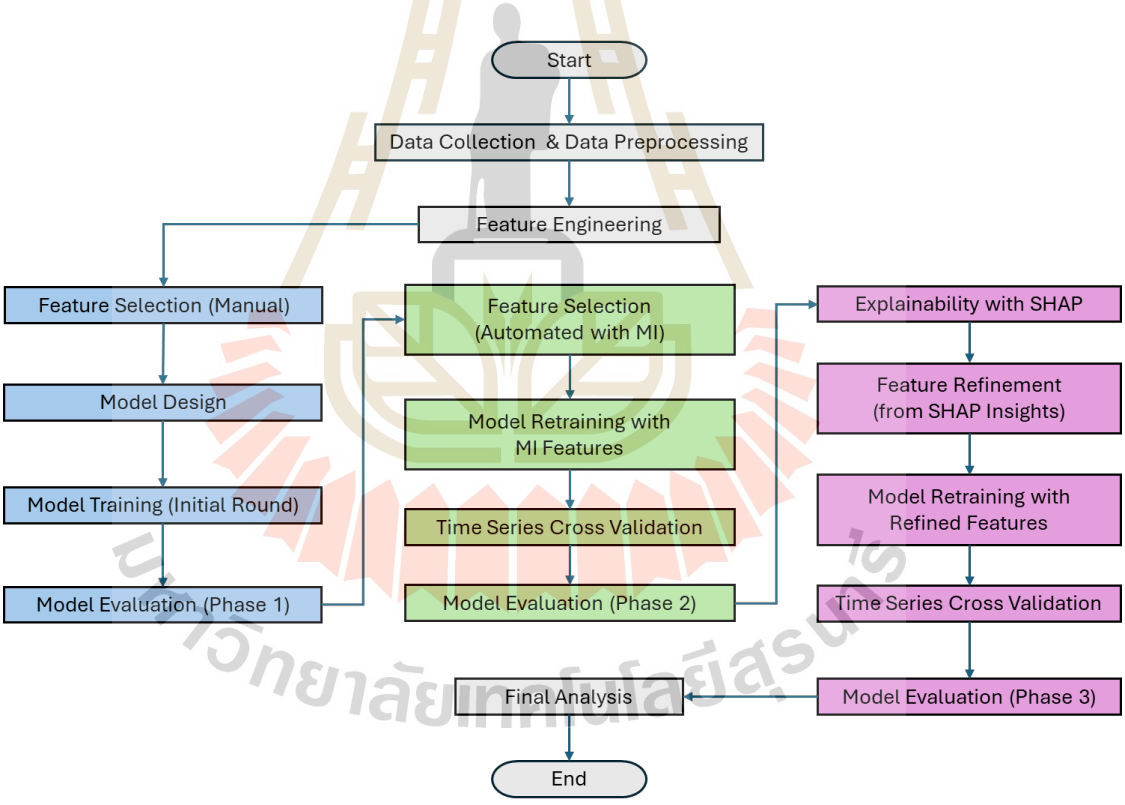


Figure 4.4 Process Description

4.9 Load Forecasting Framework

Electrical load forecasting requires the preparation of essential data, including aggregation and collection from reliable sources in the forecasting area, focusing on actual electricity load data from meters. This data is then fed into feature engineering. Two types of data are used: feature generation, which converts raw data into numbers that are easier for the model to understand and allows the model to better learn about data differences, and external feature data, which learns about data trends. This process takes place during the data collection and preparation process.

The next step in the simulation process is automatic feature selection using MI to assess the nonlinear relationship between each input variable and the target load. This helps preserve features that are important for forecasting. This reduces data complexity while retaining the most important features. The data is then grouped into three groups for training, validation, and testing, based on proportions. The resulting data is then fed into the model by designing the model, setting various parameters, and training deep learning models in various categories: Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. After training the models on the same dataset for all models, then, in the results and analysis process, the performance is evaluated using standard metrics: root means square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

To enhance the transparency and clarity of the black-box model, the SHAP tool is used to guide decision-making on feature prioritization. This tool identifies the lowest and highest priority features for the final pre-load forecast. Figure 4.5 shows the entire forecasting process. The first step is the data collection and preparation process. The next step is the simulation process. The last step is the results and analysis process.



Figure 4.5 The Complete Forecasting Framework

4.10 Simulation Results and Discussion with SUT's Campus Data

The electrical load forecast from all the data obtained and the tools mentioned above will be used in simulation using the Python 3 program to show the results comprehensively and explain the accuracy and reliability of the model in the area and context of Suranaree University of Technology. It is a 1-day-ahead load forecast using historical data for 1 year of forecasting in 2023. The details are presented in this section. Using Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models.

4.10.1 Implementation and Experimental Setup

The settings for each model are shown in Table 4.1. All models use similar parameters, namely the Dropout Rate parameter with a value of 0.2 to set the percentage of that unit. To prevent overfitting in the model, the learning rate is set to 0.001 to determine the speed of parameter tuning during learning. The optimizer uses Adam to automatically adjust the learning rate. The batch size, which is the number of data used to calculate the gradient per iteration of parameter tuning, is set to 32. The Epochs, which is the number of iterations to learn the whole data set, is set to 50. The loss function is used to evaluate the model error with MSE. In the different parts, each model is a layer part, which is the basic structure of the model that defines data processing. The DNN model has a Flatten Layer to transform the data and feed it into a Dense Layer, with the number of units in each Dense Layer having the units or the complexity of learning in each layer as 128, 64, and 1. The CNN-LSTM model has Conv1D filters, or the number of filters, as 64. The kernel size is the length of the

repeated pattern detection filter with the value of 3. The pool size for reducing the dimensionality is set to 2. The LSTM Units are set to 100 for learning the temporal relationship after Conv1D has extracted features and has a Dense value of 1 in the model part. The Attention-LSTM has an Attention Layer that prioritizes the timesteps that affect the prediction and returns 100 LSTM Units with a Dense of 1. In the LSTM model, the LSTM Unit Capacity is 100, and the Bi-LSTM model has a Bidirectional Wrapper to process two directions and uses 100 LSTM Units in each direction with a Dense of 1.

Table 4.1 Parameter Setting

Model	Parameters Layer
DNN	Flatten, Dense (128), Dense (64), Dense (1), and Units: 128, 64, 1
CNN-LSTM	Conv1D (64), kernel size (3), MaxPooling1D (pool size=2), Dense (1) and LSTM (100)
Attention-LSTM	LSTM (100), return sequences (True), Dense (1) and Attention
LSTM	LSTM (100) and Dense (1)
Bi-LSTM	Bidirectional (LSTM(100)) and Dense(1)

The selection of the appropriate hyperparameter values for each model was performed using a trial-and-error method. The parameters were adjusted within a specified range and then evaluated by evaluating the minimum MAPE value displayed when each parameter was adjusted within each range. Hyperparameter units were set in a range of 64 to 128, with a best value of 100. The dropout rate was set in a range of 0.1 to 0.3, with the best value of 0.2. The learning rate was set in a range of 0.005 to 0.0005, with a best value of 0.001. Timesteps were set in a range of 4 to 12, with a best value of 8. Batch size was set in a range of 16 to 64, with a best value of 32. Early stopping was set in a range of 5 to 20, with a best value of 10. As shown in Table 4.2, these hyperparameter sets were then used to simulate the results of each model. The details of the parameter adjustment in the Bi-LSTM model, evaluated using the MAPE

indicator, show a minimum value of 17.53% resulting in the optimal parameter range shown in Table 4.3. Example code for simulating the Bi-LSTM model is shown in Appendix B.

Table 4.2 Selecting the appropriate hyperparameter

Hyperparameter	Adjustment range	Best parameter values
Units	[64 128]	100
Dropout Rate	[0.1 0.3]	0.2
Learning Rate	[0.0005 0.005]	0.001
Timesteps	[4 12]	8
Batch Size	[16 64]	32
Early Stopping	[5 20]	10

Table 4.3 Details of parameter adjustment via indicators

Case	Units	Dropout Rate	Learning Rate	Timesteps	Batch Size	Early Stopping	MAPE (%)
1	64	0.2	0.001	8	32	10	21.28
2	128	0.2	0.001	8	32	10	19.69
3	100	0.1	0.001	8	32	10	18.64
4	100	0.3	0.001	8	32	10	21.56
5	100	0.2	0.005	8	32	10	19.95
6	100	0.2	0.0005	8	32	10	19.98
7	100	0.2	0.001	4	32	10	21.01
8	100	0.2	0.001	12	32	10	18.83
9	100	0.2	0.001	8	16	10	19.47
10	100	0.2	0.001	8	64	10	17.72
11	100	0.2	0.001	8	32	5	21.04
12	100	0.2	0.001	8	32	20	19.52
13	**100**	**0.2**	**0.001**	**8**	**32**	**10**	**17.53**

4.10.2 Mutual Information Scores of SUT Data

To prioritize the feature data before importing the model with the MI tool, the relationship of the features and the target electrical load dip data groups is measured as both linear and non-linear data. It shows the score ranking of each feature. The Lag Load feature has an MI score of 1.198, the Yearly Hours feature has an MI score of 0.976, the Daytime and Nighttime feature has an MI score of 0.405, the Ambient Temperature feature has an MI score of 0.378, the Dew Point feature has an MI score of 0.241, the Weekdays and Weekend feature has an MI score of 0.055, the Compensatory Holiday feature has an MI score of 0.018, the Long Weekend feature has an MI score of 0.014, and the Quarter-Hourly Interval feature has an MI score of 0.002, as shown in Fig. 4.6 The introduction of lag load as a feature tends to have a significant relationship between the input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast value. The behaviour of electricity users today and yesterday is similar. The features that are tested with high scores are the Yearly Hours feature, which shows the time position that repeats patterns throughout the year, showing the relationship with the forecast value in a high proportion. These are 2 features that have significantly higher scores than other models. As for other features, the scores are closer to zero and are less relevant to the forecast.

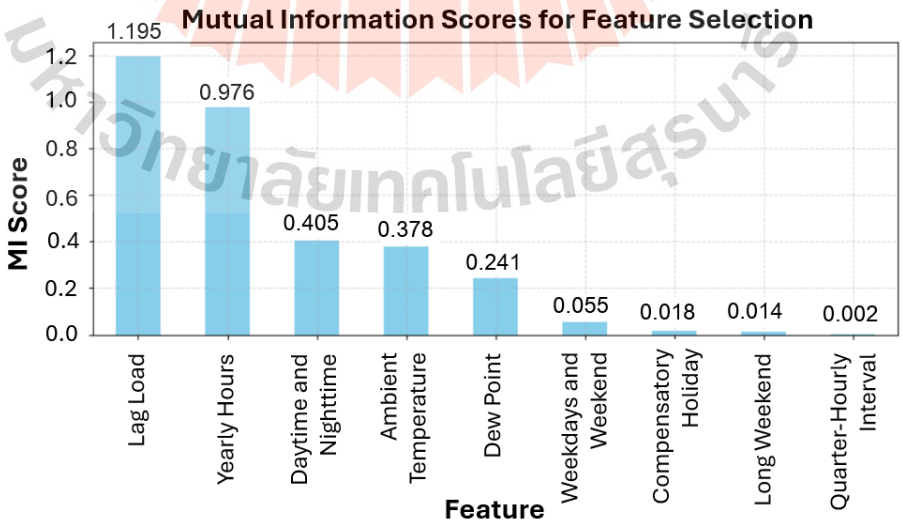


Figure 4.6 Mutual Information Scores of SUT Data

4.10.3 Daily Load Forecasting Performance

Table 4.4 Performance of Each Model on 10 Test Sets

Models	Important dates		Metric Statistic		
	Minimum	Maximum	Metric	Min.	Max.
Moving average	2023-08-01	2023-08-03	MAE (kW)	68.60	675.72
	2023-05-01	2023-08-03	RMSE (kW)	87.39	945.33
	2023-12-27	2023-09-10	MAPE (%)	5.78	78.34
Attention-LSTM	2023-04-16	2023-12-25	MAE (kW)	73.10	385.39
	2023-04-16	2023-12-25	RMSE (kW)	114.97	563.92
	2023-03-01	2023-09-10	MAPE (%)	5.99	35.40
DNN	2023-03-02	2023-12-25	MAE (kW)	97.47	403.80
	2023-03-02	2023-12-25	RMSE (kW)	122.82	615.58
	2023-09-04	2023-09-10	MAPE (%)	10.16	37.38
CNN-LSTM	2023-03-02	2023-05-02	MAE (kW)	68.33	368.25
	2023-03-02	2023-05-02	RMSE (kW)	90.55	556.91
	2023-03-02	2023-09-10	MAPE (%)	7.21	39.63
LSTM	2023-03-02	2023-05-02	MAE (kW)	83.60	362.97
	2023-03-02	2023-05-02	RMSE (kW)	105.69	528.20
	2023-09-03	2023-09-10	MAPE (%)	9.82	33.84
Bi-LSTM	2023-03-02	2023-12-25	MAE (kW)	**46.75**	373.37
	2023-03-02	2023-12-25	RMSE (kW)	**68.35**	566.79
	2023-03-02	2023-09-10	MAPE (%)	**4.82**	38.74
Bi-LSTM (Top 3 Features by SHAP)	2023-12-27	2023-09-11	MAE (kW)	63.52	394.71
	2023-03-03	2023-09-11	RMSE (kW)	90.11	612.98
	2023-03-02	2023-09-10	MAPE (%)	4.92	36.79
Bi-LSTM (Top 3 Features by MI)	2023-03-03	2023-08-03	MAE (kW)	103.28	469.44
	2023-03-03	2023-09-10	RMSE (kW)	141.87	722.92
	2023-03-02	2023-09-10	MAPE (%)	7.51	63.60

One-day-ahead electrical load forecasting at Suranaree University of Technology can demonstrate the performance of each of the six models through three metrics, compiled into statistical metrics that show the highest, lowest, and average values of each metric from ten test data sets. Important days in the forecast, including the days showing the lowest and highest values of each metric, are also included in the test data set for the model. It is shown in Table 4.4. The forecasting results show

that the moving average forecasting method gives the highest value of all 3 indicators from the tested data set. MAE has a value of 675.72 kW, and RMSE has a value of 945.33 kW on 2023-08-03, which is a date in the set of semesters start dates, as shown in Fig. 4.7 And MAPE has a value of 78.34% on 2023-09-10, which is a date in the rainy season set, as shown in Fig. 4.8 The forecasting method with the Bi-LSTM model gives the lowest values of MAE, RMSE, and MAPE indicators from the tested data set, with values of 46.75 kW, 68.35 kW, and 4.82%, respectively, which are a date in the exam set, as shown in Fig. 4.9 The figures show the load forecast for 1 day in advance for 3 consecutive days. Each model is graphed compared to the actual load graph to show the forecast trend.

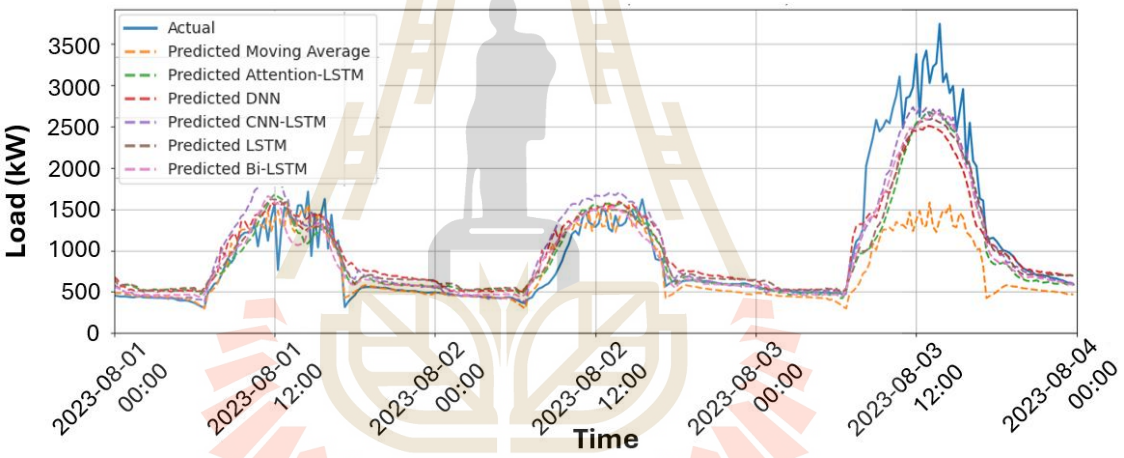


Figure 4.7 The Lowest-Performing Forecast Results in the Initial Semester Test Set

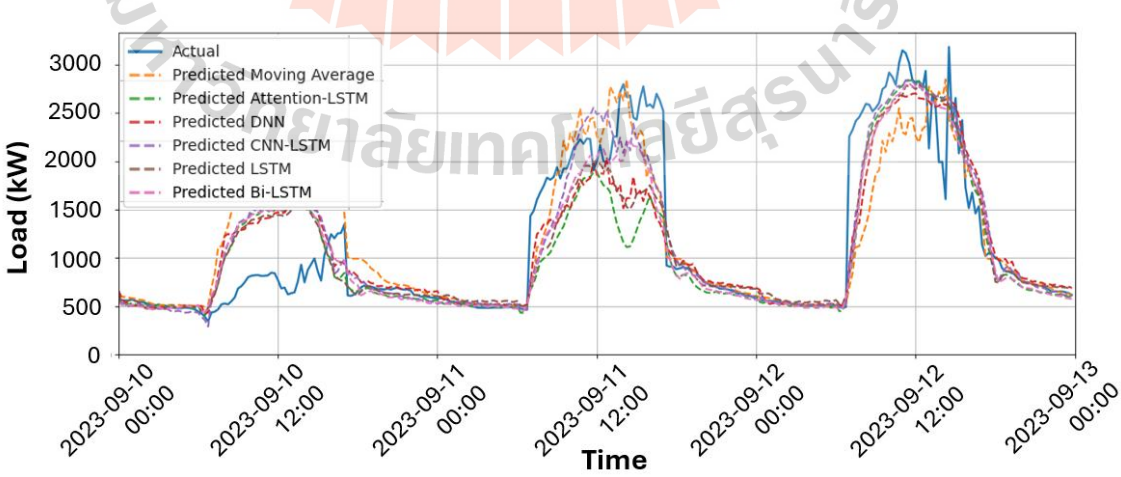


Figure 4.8 The Lowest-Performing Forecast Results in the Rainy Season Test Set

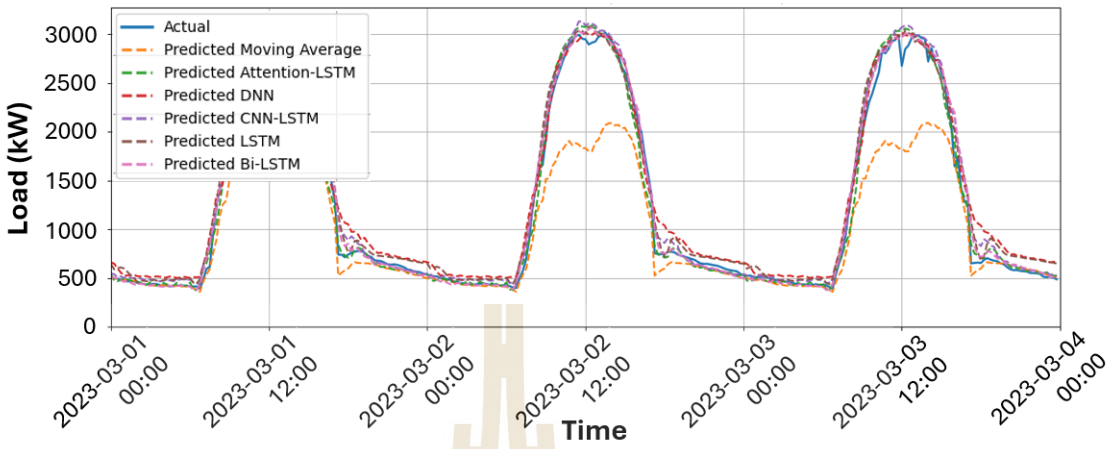


Figure 4.9 The Best-Performing Forecasting Results in the Test Set during the Exam Period

Since the moving average model relies on the average of historical load patterns, they must be similar at the same time point for accurate forecasting. However, data from the beginning of the semester, the rainy season, or the test data used for forecasting are volatile and unstable. This results in higher forecast error rates than all other models when analyzed for the maximum possible forecast error. The Bi-LSTM model has two layers that process historical data and predict future load trends, enabling it to effectively learn from data patterns that fluctuate with weather and time. Furthermore, the test data during exams or similar data have lag load and yearly hours features that significantly affect the relationship with the forecast at that time, accounting for a significant portion of the MI score. This factor occurs both before and after the forecast point. Therefore, the bidirectional learning method demonstrates the highest forecasting performance compared to all other models when analyzed for the maximum possible model accuracy. The Bi-LSTM model exhibits a lower average error (MAE) value, a percentage error compared to the actual value (MAPE), and a lower proportion of large errors (RMSE) than other models. The model can reduce errors at each time point, reduce forecasts with small errors, and

forecast peak load periods more accurately than other models. Therefore, the resulting model is the best-performing model in this context.

The LSTM model showed the lowest values of all three indicators as follows: MAE was 83.60 kW, RMSE was 105.69 kW, and MAPE was 9.82%. The model has a learning characteristic from the past to the present. There is a layer that remembers various values, which is different from the DNN model in that the features fed into the model are not chronologically ordered, which makes the indicator values higher than the LSTM model. MAE was 97.47 kW, RMSE was 122.82 kW, and MAPE was 10.16%. For the hybrid CNN-LSTM model, CNN was used to capture short-term data patterns, resulting in lower errors. The lowest values of all three indicators were as follows: MAE was 68.33 kW, RMSE was 90.55 kW, and MAPE was 7.21%. For the Attention-LSTM model, developed from a simple LSTM structure, it prioritizes the most effective timestep, showing an MAE of 73.10 kW, an RMSE of 114.97 kW, and an MAPE of 5.99%. Both models exhibit higher forecasting accuracy than the simple LSTM model. The moving average method yields the best indicators, with an MAE of 68.60 kW, an RMSE of 87.39 kW, and an MAPE of 5.78%. These are more effective than some of the models mentioned above. However, when examining the worst-day recordings, the model achieves higher indicators than all other models. Furthermore, the average test data also has the lowest overall performance.

In addition, the Bi-LSTM model was investigated to obtain the most three influential features from SHAP: Lag Load, Yearly Hours, Weekdays, and Weekend. The lowest values of the metrics were obtained as follows: MAE was 63.52 kW, RMSE was 90.11 kW, and MAPE was 4.92%. Compared to the same model set using all 9 features, the results show that the model using only the most important features provides acceptable forecast performance, although its performance is slightly reduced compared to using all features. The Lag Load feature has the highest weight in describing the daily electrical load characteristics. The Yearly Hours, Weekdays, and Weekend features still help the model better understand hourly and weekly patterns.

While selecting only 3 important features allows the model to perform well, using all features yields the lowest metric values and the best performance.

The Bi-LSTM model was re-simulated using the top three features identified as having the greatest influence by MI: Lag Load, Yearly Hours, Daytime, and Nighttime. The lowest metrics were MAE at 103.28 kW, RMSE at 141.87 kW, and MAPE at 7.51%. The MI methodology indicates that the Lag Load feature has a strong and consistent relationship with output throughout the year, demonstrating model continuity. The Yearly Hours feature can segment the load by time of year. The daytime and nighttime features can show more detailed time relationships, but they still don't yield results as good as using all nine features. For daily or one-day forecasting, the data structure requires more context than using only three features to allow the model to understand more complex details. Using only three features identifies the most important individual features. Operating alone or without the context of other features fails to recognize true relationships and can lead to higher forecasting errors.

4.10.4 Test Set Load Forecasting Performance

Table 4.5 presents the complete test dataset of the university load data, simulating the results for each of the six models. It also compares simulation cases using the top three most important features from the SHAP method and the MI method, showing the results of the overall test metrics. It was found that the model with the lowest MAE, RMSE, and MAPE metrics was the one selected using the top three most important features from the SHAP method, simulated with the Bi-LSTM model. The MAE was 177.90 kW, the RMSE was 310.29 kW, and the MAPE was 15.59%. These results indicate that the model can efficiently learn the structure of the electricity load data by selecting the most relevant features that truly impact the model. The SHAP method selects features that directly affect the model's output, thus eliminating some less relevant features. Furthermore, the memory of the Bi-LSTM

model is not used for unnecessary features; instead, more weight is given to time-related features, as the model is sensitive to temporal dependency. This results in lower overall metric values compared to other models and methods. The graphs showing the complete test suite for each model are presented in Appendix C.

Table 4.5 Overall Test Set Performance Metrics

Model	Test Set		
	MAE (kW)	RMSE (kW)	MAPE (%)
Attention-LSTM	183.72	323.18	16.68
Bi-LSTM	184.05	321.23	17.53
CNN-LSTM	195.24	332.14	19.21
LSTM	200.78	326.22	20.47
Moving Average	260.56	465.05	20.58
DNN	207.97	330.07	18.23
Bi-LSTM (Top 3 Features of SHAP)	**177.91**	**310.29**	**15.59**
Bi-LSTM (Top 3 Features of MI)	221.87	383.06	19.77

4.10.5 SHAP-Based Feature Importance Analysis

The SHAP tool was imported to describe the importance of features that affect the decision of the Bi-LSTM model, respectively, as shown in Fig. 4.10 The results of the remaining models are shown in the Appendix D. The x-axis shows the SHAP value, with a positive value indicating a higher prediction value. The y-axis sorts the features by cluster importance from highest to lowest, and the color of the dots indicates the actual feature value, with red having a high value and blue having a low value. The top three features of the NN model with the highest importance are as follows: The Lag Load feature is an important variable that has a great impact on the model's decision because it has the highest SHAP value distribution. There are a lot of red dots to the right, meaning that if yesterday's load is higher, it will also push today's

forecast value higher. The Yearly Hours feature is distributed in clusters and has most red dots to the left, meaning that features with high values will lower the forecast value. The Weekdays and Weekend features show that the closer the SHAP value distribution is to zero, the less the feature's impact on the model's decision. In the non-linear features, the SHAP values are widely distributed, with both positive and negative values and a mixture of red and blue dots at the same point. The moving average method is a basic statistical method for forecasting using features Lag 1, Lag 2, and Lag 3 as the electrical load values for the past 1, 2, and 3 days, respectively. The time-related data, namely the Hour and Minute features, are the time data for all 3 days in every hour and minute at the same time of each day. Feature Lag 1 is the feature with the most dispersion and has the most impact on the model's decision.

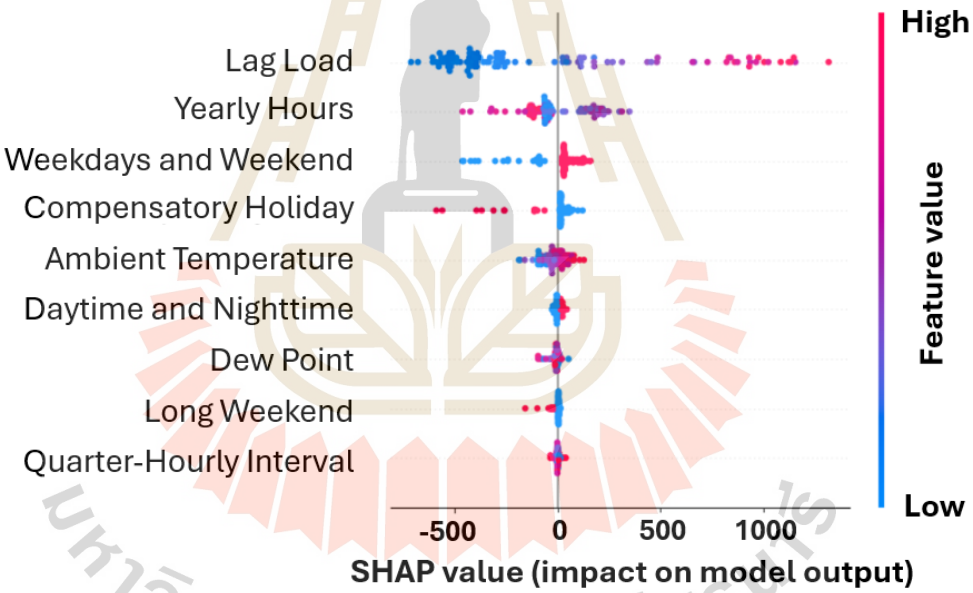


Figure 4.10 SHAP Value of Each Model: Bi-LSTM

4.10.6 Comparative Analysis of Feature Importance Models and Methods

The importance of ranking of features of NN models by SHAP and MI, where model A is LSTM, B is Bi-LSTM, C is CNN-LSTM, D is Attention-LSTM, E is DNN, and F is MI. From Table 4.6, number 1 shows the ranking of the most important features, and the higher the ranking number, the importance will decrease until

number 9, which has the lowest feature importance to the model. The Bi-LSTM model forecasts have the highest accuracy. The ranking of features from the highest to the lowest is shown as follows: Lag Load, Yearly Hours, Weekdays and Weekend, Compensatory Holiday, Ambient Temperature, Daytime and Nighttime, Dew Point, Long Weekend, and Quarter-Hourly Interval, respectively, were compared for all 6 models. The Bi-LSTM and LSTM models are models that learn the time sequence of data. The 4th feature is Compensatory Holiday, which shows the compensatory holidays of the university reflect the overall electricity consumption behavior. There are red and blue dots clearly separated from the blue ones. The model can recognize clear patterns with this data and associate it with the load drop. This differs from the fifth-ranked feature, ambient temperature. Since alternating points in the same location have a non-linear relationship, fluctuations can make it more difficult for models to capture these patterns.

Table 4.6 Feature Importance Ranking of NN Model by SHAP and MI

Feature	Feature importance ranking					
	A	B	C	D	E	F
Lag Load	1	1	1	1	1	1
Yearly Hours	2	2	2	2	2	2
Weekdays and Weekend	3	3	3	3	3	6
Compensatory Holiday	4	4	5	5	5	7
Ambient Temperature	5	5	4	4	4	4
Daytime and Nighttime	9	6	7	8	6	3
Dew Point	6	7	6	6	7	5
Long Weekend	8	8	8	7	8	8
Quarter-Hourly Interval	7	9	9	9	9	9

A CNN-LSTM model, which captures short-term and long-term patterns well, or an Attention-LSTM model that emphasizes time steps, will better understand this feature. Based on feature priorities, CNN-LSTM and Attention-LSTM models prioritize ambient temperature over compensatory holidays. DNN models can easily learn by finding association weights, do not remember event sequences, and process data without considering time sequences. The SHAP value of the Ambient Temperature feature, used to describe load changes, is higher than that of the Compensatory Holiday feature, which relies on time sequences and previous behaviors. The Quarter-Hourly Interval feature, also a later feature, is considered a fundamental feature, as it does not directly correlate with load and is a simple feature for the model to learn. From the simulation results, it was found that the MI method gives more importance to the Daytime and Nighttime, Ambient Temperature, and Dew Point features of each model than the SHAP method because these features have more raw data statistics related to the forecasting target, which reflects the actual behavior of electrical loads without learning the model mechanism and not understanding the relationship between the features like SHAP. When using SHAP, the importance is reduced. For Weekdays and Weekend and Compensatory Holiday features, which have lower MI scores, SHAP gives higher importance to both features. This is because the model learns the effect of these features through the more important Lag Load feature, not just learning the features alone. Using SHAP, one can learn features with complex relationships better.

The third feature of the MI method, Daytime and Nighttime, is shown to be highly correlated with load. However, compared to the SHAP method, this feature is ranked low due to the model's inability to extract patterns well. However, the key features, Lag Load and Yearly Hours, ranked first and second in both methods, showed consistent results, indicating that this feature set can definitely improve accuracy and efficiency. Using load data, the MI method screens out potentially important features related to electrical loads, screening relevant and related data before importing into

the model and isolating truly irrelevant features. The trend in importance of features is evident, eliminating bias from the model for optimal forecasting performance. However, since it is not yet possible to determine how features affect model decisions, SHAP values are used at each data point in real-world situations to explain the relationship between features and model decisions. This ensures that the model is indeed using those features and provides a deeper explanation of model behavior, demonstrating the importance of features that directly influence the model. Furthermore, under different load conditions, the direction of their impact on forecasting can be identified. Regarding the reliability of black-box models, this can be explained.

4.11 Simulation Result and Discussion with ISO New England Data

In electricity load forecasting, understanding the context of the data and the influence of features on the results of various models is crucial. Improving forecasting performance and model description are key issues, as are the repeatability and reliability of the research framework. Therefore, electricity load forecasting was simulated using hourly electricity demand data from January 3, 2003, to December 31, 2014, compiled and published by ISO New England (ISO-NE), an independent agency responsible for regulating and managing electricity systems in the New England region of the United States, as an example to illustrate the analysis of the models and methods proposed above. Using Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models.

4.11.1 Mutual Information Scores of ISO-NE Data

Prioritizing feature data before importing them into the model using the MI tool measures the correlation between features and dip data sets of electricity demand, compiled and published by ISO New England (ISO-NE) of the United States. Both linear and non-linear data are used, showing the ranking scores of each feature. The Lag Load feature has an MI score of 0.977, Yearly Hours 0.487, Ambient

Temperature 0.386, Month 0.171, Weekdays 0.0516, Year 0.023, and Day 0.011, as shown in Fig. 4.11. It can be seen that including lag load as a feature shows a significant correlation between input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast values, with electricity user behavior today and yesterday being very similar. The features with the lowest scores are yearly hours, which show recurring time patterns throughout the year and demonstrate a high correlation with the predicted values. These two features show significantly higher scores than other models. Features with scores close to 0 show decreasing correlation with the predictions in descending order.

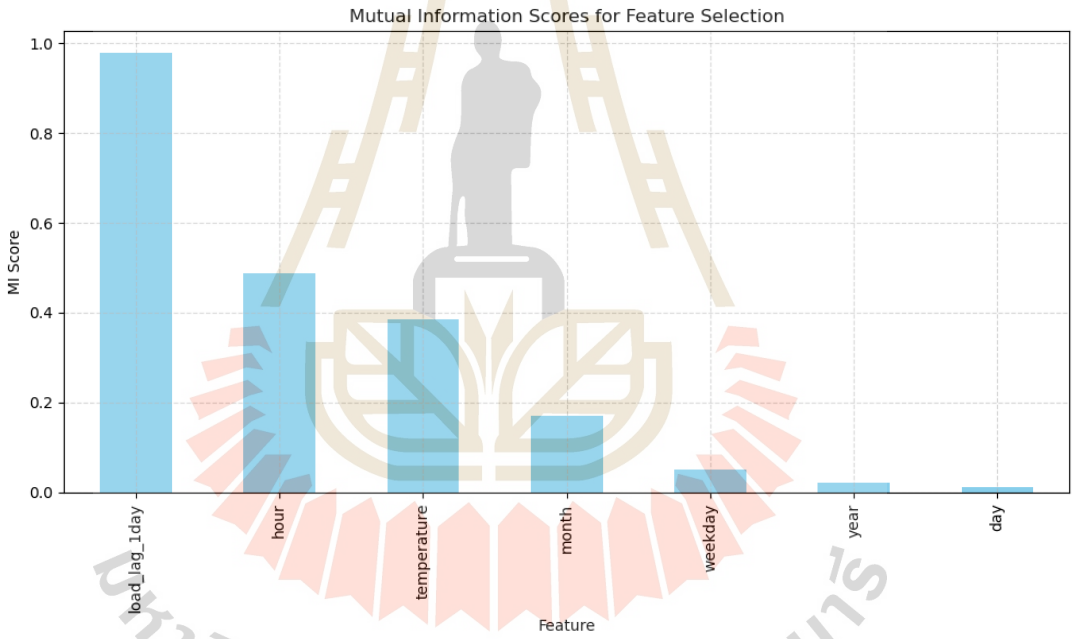


Figure 4.11 Mutual Information Scores of ISO-NE Data

4.11.2 Daily Load Forecasting Performance

The electricity load forecasts compiled and published by ISO New England (ISO-NE) one day in advance can show the performance of each of the six models through three indicators. These are compiled as statistical metrics indicating the lowest values for each model from the tested data. Key forecast dates, including

those showing the lowest values, are also included in the test dataset, as shown in Table 4.7. The forecast results show that the CNN-LSTM method yielded the lowest values for all three indicators from the tested dataset, with MAE at 89.08 kW, RMSE at 118.94 kW, and MAPE at 0.67% on October 9, 2014, making it the most accurate model.

Table 4.7 The best result for each model in daily forecasting

Models	Important dates	Metric Statistic		
	Date of Minimum	MAE (kW)	RMSE (kW)	MAPE (%)
Moving Average	2014-09-25	98.40278	142.3815	0.71
Attention-LSTM	2014-06-10	108.78	137.65	0.74
DNN	2014-07-19	238.51	281.87	1.75
LSTM	2014-10-21	90.73	126.32	0.69
Bi-LSTM	2014-06-05	96.84	123.52	0.70
CNN-LSTM	2014-10-02	**89.08**	**118.94**	**0.67**
CNN-LSTM (Top 3 features SHAP and MI)	2014-10-09	148.25	203.14	1.13

Fig. 4.12 shows a flat graph comparing actual load and other models to illustrate forecast trends for that day. The figure shows three consecutive 1-day load forecasts. The LSTM model yielded the lowest values: MAE 90.73 kW, RMSE 126.32 kW, and MAPE 0.69% on October 21, 2014. The Bi-LSTM model yielded the lowest values: MAE 96.84 kW, RMSE 123.52 kW, and MAPE 0.70% on June 5, 2014. The Moving Average model yielded the lowest values: MAE 98.40 kW, RMSE 142.38 kW, and MAPE 0.71% on date missing. On September 25, 2014, the forecasting method using the Attention-LSTM model yielded the following lowest metric values: MAE at 108.78 kW, RMSE at 137.65 kW, and MAPE at 0.74% on June 10, 2014. The forecasting method using the

DNN model yielded the following lowest metric values: MAE at 238.51 kW, RMSE at 281.87 kW, and MAPE at 1.75% on July 19, 2014. The models display the metric results from lowest to highest values, respectively.

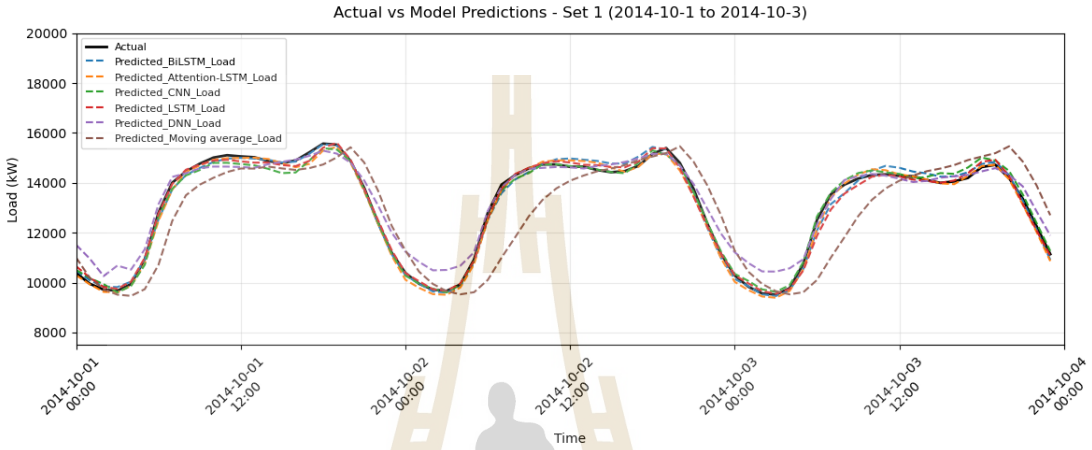


Figure 4.12 Comparison of the best forecasting results of the CNN-LSTM model

The results show that the CNN-LSTM hybrid model is the most accurate for 1-day forecasting. This is because the forecasted data contains recurring daily load changes over short forecast periods. The CNN model analyzes the similarity of these short-term patterns before passing them to the LSTM model to learn longer-term patterns, resulting in the lowest forecast metric. The LSTM model, which is less accurate, primarily learns from time-series memory and performs well on relatively simple datasets. Bi-LSTM models typically learn both past and future patterns, indicating that future pattern learning is not very helpful for this type of dataset. Moving average models, acting as baselines, only learn from historical electricity load data and therefore cannot understand more complex data. Attention-LSTM models can learn from time series datasets that span many steps ahead and are more complex than basic models. However, using them for 1-day forecasting on a fixed-pattern dataset would make the model too complex and unsuitable for the data. The creation of excess parameters leads to worse results compared to other models, and basic DNN

models show the least accurate results because they lack a temporal understanding of forecasting. At each time point, the model does not understand the temporal relationships between features, which is a crucial issue for electricity load forecasting. When there are many parameters used in forecasting, the model is prone to overfitting.

Furthermore, a re-simulation of the most accurate CNN-LSTM model was performed using the top three most influential features identified from SHAP and MI: Lag Load, Yearly Hours, and Ambient Temperature, respectively. Analysis of the data and model, along with statistical methods, indicated that historical load shows the most significant pattern in electricity consumption and has the greatest influence on usage. This is followed by temporal features reflecting important daily patterns, and weather factors tend to reflect electricity usage based on temperature. The lowest values were recorded on October 9, 2014: MAE at 148.25 kW, RMSE at 203.14 kW, and MAPE at 1.13%. The simulation results showed that while the three identified features are highly significant, using only these three results in less accurate forecasting compared to using all features. This is because the model learning process requires diverse datasets and contexts to effectively identify relationships and achieve optimal performance.

4.11.3 Test Set Load Forecasting Performance

Table 4.8 presents the complete test dataset, simulating each of the six models and comparing cases simulating the top three most important features from the SHAP and MI methods with the CNN-LSTM model. The results show the indicators of the entire test set. The model with the lowest MAE is CNN-LSTM, with an MAE value of 333.99 kW. The model with the lowest RMSE is Attention-LSTM, with an RMSE value of 475.15 kW. And the model with the lowest MAPE is CNN-LSTM, with a MAPE value of 2.29%. These results indicate that the CNN-LSTM model is the most stable and accurate, predicting closest to the true values with low MAE (low mean error per data point) and low MAPE (low percentage error across the load range). The Attention-LSTM

model has the lowest RMSE, indicating its ability to control errors occurring during high load periods. This demonstrates the good predictive ability of the entire test dataset during peak load periods. The graphs showing the complete test suite for each model are presented in Appendix C.

Table 4.8 Overall Test Set Performance Metrics of ISO-NE Data

Models	Test Set		
	MAE (kW)	RMSE (kW)	MAPE (%)
Attention-LSTM	334.5389	**475.1453**	2.32
Bi-LSTM	368.1364	534.2110	2.50
CNN-LSTM	**333.9936**	497.3806	**2.29**
LSTM	358.5621	516.5337	2.45
Moving Average	1129.35	1601.82	7.67
DNN	611.4789	824.5546	4.22
CNN-LSTM (Top 3 features SHAP and MI)	625.95	848.3157	4.34

4.11.4 SHAP-Based Feature Importance Analysis

SHAP is used to explain the importance of features affecting the decision-making of Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models, respectively, as shown in Appendix D. The top three most important features in NN models are Lag Load, shown in Fig. 4.13, a key variable significantly impacting model decision-making due to its largest SHAP distribution (shown with red dots close to the right and left, indicating that higher yesterday's load will raise today's forecast, and lower yesterday's load will lower today's forecast); Yearly Hours, which is clustered with purple dots concentrated on the right, suggesting that mid-range values improve forecasts; and Ambient Temperature, where the purple dots are clustered on the left in SHAP distribution, meaning that mid-range values (neither too high nor too low) decrease forecasts. The weekday, month, day, and year features show a SHAP

distribution closer to zero. The impact of features on model decision-making decreases with increasing frequency. The moving average method is a basic method that uses statistical values for forecasting, employing Lag 1, Lag 2, and Lag 3 features as historical electrical load data for the past 1, 2, and 3 days, respectively. The time-related data, the Hour feature, represents hourly data for all three days at the same time each day. The Hour feature is the most widely distributed and has the greatest impact on model decision-making.

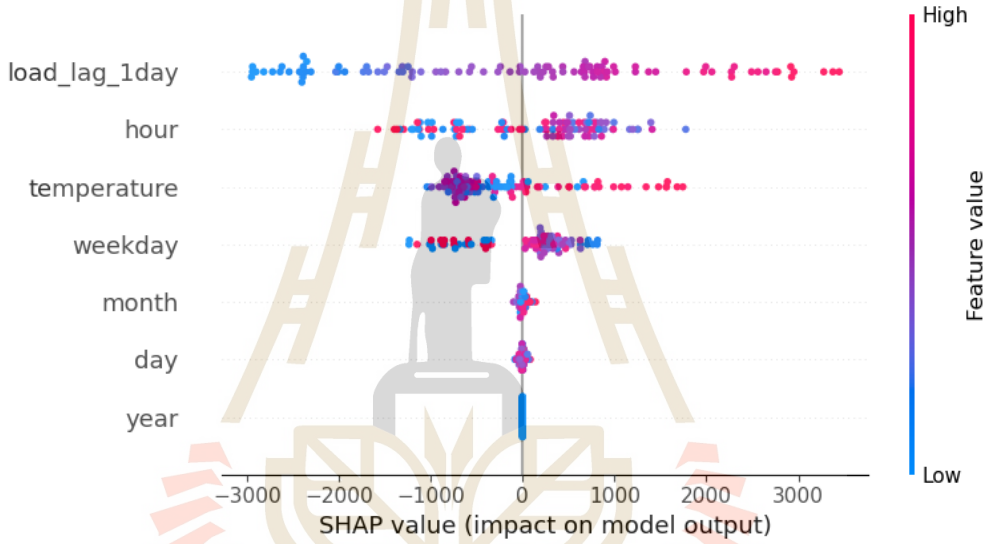


Figure 4.13 SHAP Value of Each Model of ISO-NE Data: CNN-LSTM

4.11.5 Comparative Analysis of Feature Importance Models and Methods

Table 4.9 shows the priority of features from highest to lowest, with increasing priority as the number increases, down to feature number 9, which has the lowest importance to the model. CNN-LSTM models show the highest forecasting accuracy. Based on SHAP analysis, the priority of features from highest to lowest is as follows: Lag Load, Yearly Hours, Ambient Temperature, Weekday, Month, Day, and Year. A comparison of the six models is shown. LSTM and Attention-LSTM models have similar priorities. Bi-LSTM models have a similar priority, with features 2 (temperature) and 3 (hourly) swapped. Because the models learn in two directions within the same

sequence, there is temporal continuity, and the temperature feature exhibits this continuity, giving it a higher priority in Bi-LSTM. The hourly feature, however, does not translate according to time but simply indicates the position of a 24-hour cycle. DNN models have a similar priority, with features 5 (Day) and 6 (Month) swapped, as temporal continuity is a factor. The model does not recognize the time sequence and directly affects the load. However, the Day feature clearly impacts electricity user behavior, making it easy for the model to grasp the relationship, thus ranking higher in DNN models.

Table 4.9 Feature Importance Ranking of ISO-NE Data

Feature	Feature importance ranking of NN model by SHAP and MI					
	LSTM	Bi-LSTM	CNN-LSTM	Attention-LSTM	DNN	MI
Lag Load	1	1	1	1	1	1
Yearly Hours	2	3	2	2	2	2
Ambient Temperature	3	2	3	3	3	3
Weekday	4	4	4	4	4	5
Month	5	5	5	5	6	4
Day	6	6	6	6	5	7
Year	7	7	7	7	7	6

Based on the MI analysis, the feature priority is similar to that of the CNN-LSTM model by SHAP, but with some positional changes. Feature position 4 represents the Month feature, and feature position 5 represents the Weekday feature. Feature positions 6 represent the Year feature, and 7 represents the Day feature. The MI method views the overall statistical relationship of the data, prioritizing features that result in the lowest forecasting error. The Month feature highlights different seasons, and since the MI method considers the overall year and therefore has lower

error margins, it is more important than the Weekday feature, which recurs over short periods. Similarly, the Year and Day features are given higher priority by the MI method because they can better separate load groups over longer periods.

4.12 Summary of Results

This research proposes the development of forecasting techniques using deep learning. The accuracy of the models was compared to underlying neural network (NN) models. Bi-LSTM and CNN-LSTM models showed the highest accuracy and performance for forecasting daily electricity consumption. Data collected, after cleaning and feature generation, was transformed into features ready for input into forecasting models. Electricity consumption forecasts were most consistent with actual values. Overall, the Bi-LSTM model with feature selection using the SHAP method, Attention-LSTM, and CNN-LSTM provided the most accurate forecasts and also yielded the lowest average metrics among all tested datasets, indicating the models' potential for forecasting in various contexts.

Furthermore, the model reliability investigation indicates that transparent feature analysis and model description using MI can explain the relationships between raw data and electricity loads and filter static data for inspection before inputting the model. This reduces the number of features required for model input and reduces training time. SHAP provides in-depth model descriptions, demonstrating the actual use of the model's features, understanding complex features, and increasing model reliability because all features have underlying relationships that can be demonstrated. Using this combined approach, comprehensive coverage is achieved. It provides accurate forecasting results and can also provide causal explanations.

CHAPTER V
CONCLUSION

5.1 Conclusion

This study proposes a method for developing a one-day advance electricity load forecasting model using data from Suranaree University of Technology. It employs deep learning techniques, including Day-Aware Data Arrangement (DGA), to compare deep learning models such as LSTM, GRU, Bi-LSTM, and Bi-GRU. The results indicate that LSTM and Bi-LSTM tend to provide the most accurate forecasts, with GRU showing the fastest simulation time in initial testing. Subsequently, selected models—Deep Learning, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM—were further selected for comparison with baseline models to improve accuracy and reliability. This was achieved through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis, optimized through data collection and feature engineering tailored to the context of the university's integrated solar power generation capacity. Using three accuracy indices—Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE)—the model from the NN group, the Bi-LSTM model using the SHAP 3 feature selection method for the top 3 most important features, provided the best overall forecasting performance from all test datasets. The Bi-LSTM model was also the most accurate, yielding the lowest metric values for daily forecasts compared to all other models. Feature selection that aligns with the model's learning mechanism has a greater impact on performance than considering statistical relationships alone. However, experimental results also indicate that using only the top three features does not always yield the best results. In highly complex datasets, such as those with holidays, seasons, and weather influences, using multiple features allows the model to learn the data context more comprehensively. Another dataset used for simulation, the ISO New England dataset, showed that the Attention-LSTM model performed best in forecasting electricity load across all test datasets. With the

lowest RMSE and MAE/MAPE values, CNN-LSTM is the most suitable model for this type of dataset, exhibiting the best overall forecasting performance. The CNN-LSTM model also shows the highest accuracy in daily pre-load forecasting. Time-sequence models, particularly Bi-LSTM and CNN-LSTM, significantly outperform time-structure insensitive models, reflecting the fact that the ability to learn past and future relationships in time-sequence data is a crucial factor in electricity load forecasting accuracy.

Analysis of the features that have the greatest impact and influence on electricity load forecasting using both MI statistical analysis and SHAP principles revealed that the lag load feature, or the value of electricity load over the past 1 day, is the feature most correlated with the forecasting target and has the greatest impact on the model's decision-making. This characteristic was consistent across both datasets. MI, as a statistical measure, can capture nonlinear relationships and is easy to use, but it's unclear how the feature is actually utilized by the model. It's suitable for initial feature screening and as a statistical indicator of how much a feature shares information with the target variable, independent of any model. In contrast, SHAP analysis can show the relationship between data and the model, allowing for the selection of necessary features and reducing forecasting errors. It's suitable for datasets that show relatively consistent electricity load behavior, clear time-repeating patterns, and are driven by only a few key features, leading to more accurate forecasts. It's ideal for in-depth feature selection. However, the priority of features obtained from MI and SHAP differs due to their different objectives.

The performance of electricity load forecasting does not depend solely on the model or feature selection method but also on the consistency between the data structure, model, and shared feature set. The findings can be applied to the design of electricity load forecasting systems in the context of smart energy systems and provide guidance for selecting appropriate models and features for future time-scheduling forecasting tasks.

5.2 Suggestions

Based on the findings of this study, several recommendations can be proposed to enhance the applicability and robustness of short-term electricity load forecasting models in future research. First, the incorporation of additional explanatory variables, such as electricity price signals, socioeconomic factors, and distributed energy resource penetration, may further improve forecasting accuracy and model generalization under more diverse operating conditions.

Second, although this study demonstrates the effectiveness of advanced recurrent neural network architectures combined with feature selection and interpretability techniques, future work could explore hybrid and ensemble learning strategies to leverage the complementary strengths of different model structures. The integration of probabilistic forecasting approaches is also recommended to provide uncertainty information that can support more informed decision-making in power system operation and planning.

Third, the use of advanced model interpretability techniques could be further extended to investigate dynamic feature importance across different time periods and operating scenarios. This would provide deeper insights into how electricity consumption patterns evolve over time and under varying conditions, particularly during extreme events or atypical load behaviors.

Finally, future studies may consider applying the proposed forecasting framework to larger-scale power systems or multi-region datasets with different load characteristics to further validate its robustness and generalizability. Such extensions would contribute to the development of reliable, interpretable, and practical electricity load forecasting systems for real-world applications.

REFERENCES

Abumohsen, M., Owda, A. Y., & Owda, M. (2023). Electrical load forecasting using LSTM, GRU, and RNN algorithms. Energies, 16(5), 2283. https://doi.org/10.3390/en16052283

Atef, S., & Eltawil, A. B. (2020). Assessment of stacked unidirectional and bidirectional long short-term memory networks for electricity load forecasting. Electric Power Systems Research, 189, 106489. https://doi.org/10.1016/j.epsr.2020.106489

Cordeiro-Costas, M., Villanueva, D., Eguía-Oller, P., Martínez-Comesaña, M., & Ramos, S. (2023). Load forecasting with machine learning and deep learning methods. Applied Sciences, 13(13), 7933. https://doi.org/10.3390/app13137933

Fan, S., & Hyndman, R. J. (2012). Short-term load forecasting based on a semi-parametric additive model. IEEE Transactions on Power Systems, 27(1), 134–141. https://doi.org/10.1109/TPWRS.2011.2162082

Habbak, H., Mahmoud, M., Metwally, K., Fouda, M. M., & Ibrahem, M. I. (2023). Load forecasting techniques and their applications in smart grids. Energies, 16(3), 1480. https://doi.org/10.3390/en16031480

Jang, J., Jeong, W., Kim, S., Lee, B., Lee, M., & Moon, J. (2023). RAID: Robust and interpretable daily peak load forecasting via multiple deep neural networks

and Shapley values. Sustainability, 15(8), 6951. https://doi.org/10.3390/su15086951

Koprinska, I., Rana, M., & Agelidis, V. G. (2015). Correlation and instance based feature selection for electricity load forecasting. Knowledge-Based Systems, 82, 29–40. https://doi.org/10.1016/j.knosys.2015.02.017

Lee, Y.-G., Oh, J.-Y., Kim, D., & Kim, G. (2023). SHAP value-based feature importance analysis for short-term load forecasting. Applied Science and Convergence Technology, 18, 579–588. https://doi.org/10.1007/s42835-022-01161-9

Liu, X., Yang, Z., Guo, Y., Li, Z., & Xu, X. (2024). A novel correlation feature self-assigned Kolmogorov-Arnold Networks for multi-energy load forecasting in integrated energy systems. Energy Conversion and Management, 300, 119388.

Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy usage forecasting model based on long short-term memory (LSTM) and explainable artificial intelligence (XAI). Information, 14(5), 265. https://doi.org/10.3390/info14050265

Mpawenimana, I., Pegatoquet, A., Roy, V., Rodriguez, L., & Belleudy, C. (2020). A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing. 2020 IEEE Sensors Applications Symposium (SAS), 1–6. https://doi.org/10.1109/SAS48726.2020.9220021

Neubauer, A., Brandt, S., & Kriegel, M. (2025). Explainable multi-step heating load forecasting: Using SHAP values and temporal attention mechanisms for enhanced interpretability. Energy and AI, 100480. https://doi.org/10.1016/j.egyai.2025.100480

Panapongpakorn, T., & Banjerdpongchai, D. (2019). Short-term load forecast for energy management system using neural networks with mutual information method of input selection. 2019 SICE International Symposium on Control Systems (SICE ISCS), 13–18. https://doi.org/10.23919/SICEISCS.2019.8758714

Pavlatos, C., Makris, E., Fotis, G., Vita, V., & Mladenov, V. (2023). Enhancing electrical load prediction using a bidirectional LSTM neural network. Electronics, 12(22), 4652. https://doi.org/10.3390/electronics12224652

Pirbazari, A. M., Chakravorty, A., & Rong, C. (2019). Evaluating feature selection methods for short-term load forecasting. 2019 IEEE International Conference on Big Data and Smart Computing (BigComp). https://doi.org/10.1109/BIGCOMP.2019.8679188

Prakash, S., Jalal, A. S., & Pathak, P. (2023). Forecasting COVID-19 pandemic using Prophet, LSTM, hybrid GRU-LSTM, CNN-LSTM, Bi-LSTM and Stacked-LSTM for India. 2023 6th International Conference on Information Systems and Computer Networks (ISCON), 1–6. https://doi.org/10.1109/ISCON57294.2023.10112065

Shi, H., Xu, M., & Li, R. (2018). Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Transactions on Smart Grid, 9(5), 5271–5280. https://doi.org/10.1109/TSG.2017.2686012

Sias, Q. A., Gantassi, R., & Choi, Y. (2024). Multivariate bidirectional gate recurrent unit for improving accuracy of energy prediction. ICT Express. https://doi.org/10.1016/j.icte.2024.10.002

Ullah, K., Khan, M. A., Hassan, M. A., Ullah, A., & Lee, M. Y. (2024). Short-term load forecasting: A comprehensive review and simulation study with CNN–LSTM hybrids approach. IEEE Access, 12, 111858–111881. https://doi.org/10.1109/ACCESS.2024.3440631

Vanitha, R., & Swaroopan, N. M. J. (2023). Contrast of stacked GRU-RNN and bidirectional LSTM network based process for forecasting renewable energy for smart grid. 2023 International Conference on RMKMATE, Chennai, India. https://doi.org/10.1109/RMKMATE59243.2023.10369285

Van Zyl, A., Heyns, T., & Bekker, J. (2024). Exploring the effectiveness of SHAP for feature selection in national demand forecasting. Energy Reports, 10, 152–162. https://doi.org/10.1016/j.egyr.2024.01.012

van Zyl, C., Ye, X., & Naidoo, R. (2023). Harnessing explainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP. Applied Energy, 344, 122079. https://doi.org/10.1016/j.apenergy.2023.122079

Wang, J., Yu, B., Chen, X., Dai, G., Liu, W., He, N., Zhu, P., Yin, Z., & Pan, Z. (2025). An interpretable short-term electrical load forecasting model based on SHapley Additive exPlanations—A case study in Haidian, Beijing. Electric Power Systems Research, 226, 111769. https://doi.org/10.1016/j.epsr.2025.111769

Wu, P., Huang, Z., Pian, Y., Xu, L., Li, J., & Chen, K. (2020). A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting. Journal of Advanced Transportation, 2020, 1–12. https://doi.org/10.1155/2020/8863724

มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX

มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX A

Performance of LSTM model on 12 test sets

Table A.1 Performance of LSTM Model on 12 Test Sets

Set	Data types	Date of data set	Performance of LSTM models		
			MAPE (%)	MAE (kW)	RMSE (kW)
1	Long holiday	2023-01-02	15.76	102.81	156.36
		2023-01-03	12.27	134.20	228.64
		2023-01-04	11.34	97.46	126.41
2	Weekdays during the school term	2023-02-01	24.72	397.40	765.97
		2023-02-02	12.36	180.62	283.29
		2023-02-03	24.01	280.92	424.00
3	Exam period	2023-03-01	6.34	90.21	157.10
		2023-03-02	**5.93**	83.05	145.60
		2023-03-03	10.59	126.19	192.18
4	Summer season	16.49	135.60	199.88	16.49
		8.49	68.27	102.64	8.49
		9.31	111.19	188.19	9.31
5	Weekdays during the school term	8.24	90.70	131.63	8.24
		20.35	298.40	452.52	20.35
		18.12	288.76	439.26	18.12
6	The start of the semester	2023-08-01	13.39	110.15	173.10
		2023-08-02	6.78	**59.18**	**93.66**
		2023-08-03	15.00	313.06	486.03

มหาวิทยาลัยเทคโนโลยีสุรนารี

Table A.1 Performance of LSTM Model on 12 Test Sets (continued)

Set	Data types	Date of data set	Performance of LSTM models		
			MAPE (%)	MAPE (%)	MAPE (%)
7	Compensation holiday period	2023-09-02	13.10	141.77	226.18
		2023-09-03	11.15	104.84	150.12
		2023-09-04	11.09	236.59	401.60
8	Rainy season	2023-09-10	38.89	277.84	415.75
		2023-09-11	15.71	306.48	490.29
		2023-09-12	12.21	207.54	344.20
9	Weekdays and weekends	2023-10-01	9.72	122.79	233.52
		2023-10-02	10.31	187.96	314.12
		2023-10-03	12.33	173.16	256.57
10	Exam period	2023-10-16	10.99	200.85	330.97
		2023-10-17	8.72	148.79	230.52
		2023-10-18	9.05	149.75	247.86
11	Term break	2023-11-01	13.50	201.88	318.32
		2023-11-02	12.40	191.58	301.04
		2023-11-03	19.65	229.27	350.74
12	Winter season	2023-12-25	26.53	404.89	625.17
		2023-12-26	12.48	187.36	325.77
		2023-12-27	19.21	354.07	564.38

มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX B

Implementation and Experimental Setup

B.1 Bi-LSTM model forecasting code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import TimeSeriesSplit
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import mean_absolute_error, mean_squared_error


# ---------- Helper Functions ---------- #
def mean_absolute_percentage_error_manual(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
def create_bilstm_model(input_shape, lstm_units=100, dropout_rate=0.2,
learning_rate=0.001):
    model = Sequential()
    model.add(Bidirectional(LSTM(units=lstm_units), input_shape=input_shape))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1))
    optimizer = Adam(learning_rate=learning_rate)
```

```
    model.compile(optimizer=optimizer, loss='mse')
```

B.1 Bi-LSTM model forecasting code (continued)

```
    return model
def create_sequences(X, y, timesteps):
    Xs, ys = [], []
    for i in range(len(X) - timesteps):
        Xs.append(X[i:i+timesteps])
        ys.append(y[i+timesteps])
    return np.array(Xs), np.array(ys)


# ------------------ Load Dataset ------------------ #
file_train_val = r"D:\New folder\Load forcasting 02\Data\trainNvalidation.xlsx"
file_test = r"D:\New folder\Load forcasting 02\Data\test.xlsx"
df_train_val = pd.read_excel(file_train_val)
df_test = pd.read_excel(file_test)


# ------------------ Prepare Data ------------------ #
selected_features = [
    'Yearly Hours', 'Quarter-Hourly Interval', 'Weekdays and Weekend',
    'Daytime and Nighttime', 'Compensatory Holiday', 'Long Weekend',
    'Lag_1Day_Smoothed_Load', 'Ambient Temperature', 'Relative Humidity']
features = df_train_val[selected_features].values
target = df_train_val['Pure Load '].values.reshape(-1, 1)
test_features = df_test[selected_features].values
test_target = df_test['Pure Load '].values.reshape(-1, 1)
test_time = df_test.iloc[:, 0]
```

B.1 Bi-LSTM model forecasting code (continued)

```
# Normalize
scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()
X_scaled = scaler_X.fit_transform(features)
y_scaled = scaler_y.fit_transform(target)
timesteps = 8
X_seq, y_seq = create_sequences(X_scaled, y_scaled, timesteps)

# ---------- K-Fold with TimeSeriesSplit ---------- #
tscv = TimeSeriesSplit(n_splits=5)
val_metrics = []
fold = 1
for train_idx, val_idx in tscv.split(X_seq):
    X_train, X_val = X_seq[train_idx], X_seq[val_idx]
    y_train, y_val = y_seq[train_idx], y_seq[val_idx]
    model = create_bilstm_model((timesteps, X_train.shape[2]))
    early_stop = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
    model.fit(X_train, y_train, epochs=50, batch_size=32,
              validation_data=(X_val, y_val),
              callbacks=[early_stop], verbose=0)
    y_val_pred_scaled = model.predict(X_val)
    y_val_pred = scaler_y.inverse_transform(y_val_pred_scaled)
    y_val_actual = scaler_y.inverse_transform(y_val)
    mae_val = mean_absolute_error(y_val_actual, y_val_pred)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    rmse_val = np.sqrt(mean_squared_error(y_val_actual, y_val_pred))
```

B.1 Bi-LSTM model forecasting code (continued)

```
    mape_val = mean_absolute_percentage_error_manual(y_val_actual, y_val_pred)
    val_metrics.append((mae_val, rmse_val, mape_val))
    print(f"\nFold {fold} Validation Metrics:")
    print(f"MAE: {mae_val:.4f} kW, RMSE: {rmse_val:.4f} kW, MAPE: {mape_val:.2f}%")
    fold += 1


# ------------------ Evaluate on Test Set ------------------ #
X_test_scaled = scaler_X.transform(test_features)
y_test_scaled = scaler_y.transform(test_target)
X_test_seq, y_test_seq = create_sequences(X_test_scaled, y_test_scaled, timesteps)
test_time = test_time[timesteps:]
y_test_pred_scaled = model.predict(X_test_seq)
y_test_pred = scaler_y.inverse_transform(y_test_pred_scaled)
y_test_actual = scaler_y.inverse_transform(y_test_seq)
test_df = pd.DataFrame({
    "Time": test_time.reset_index(drop=True),
    "Actual": y_test_actual.flatten(),
    "Predicted": y_test_pred.flatten()
})
test_df['Date'] = pd.to_datetime(test_df['Time']).dt.date
print("\nTest Set Metrics by Day:")
for date in test_df['Date'].unique():
    daily = test_df[test_df['Date'] == date]
    mae = mean_absolute_error(daily['Actual'], daily['Predicted'])
    rmse = np.sqrt(mean_squared_error(daily['Actual'], daily['Predicted']))
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    mape = mean_absolute_percentage_error_manual(daily['Actual'], daily['Predicted'])
```

B.1 Bi-LSTM model forecasting code (continued)

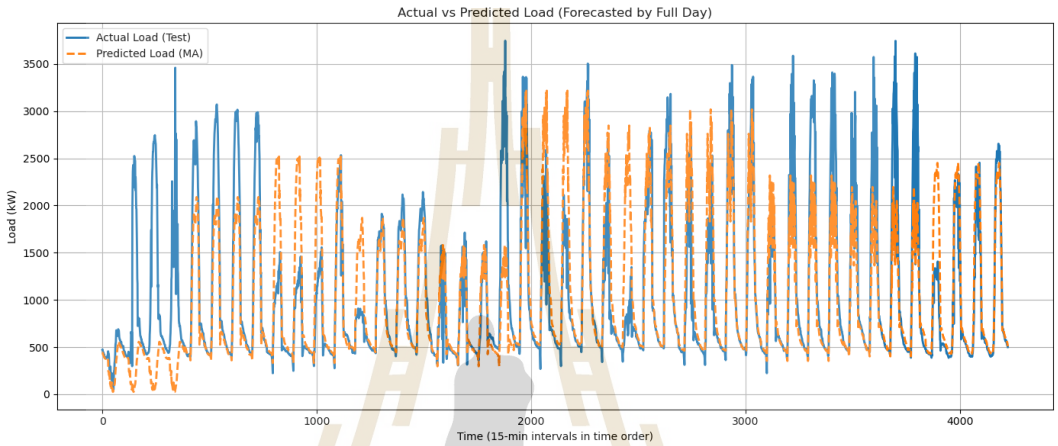
```
    print(f"{date} -> MAE: {mae:.2f} kW, RMSE: {rmse:.2f} kW, MAPE: {mape:.2f}%")

# ------------------ Plot Results ------------------ #
plt.figure(figsize=(14, 6))
plt.plot(y_val_actual, label='Actual Load (Validation)', linestyle='-', linewidth=2)
plt.plot(y_val_pred, label='Predicted Load (Validation)', linestyle='--', linewidth=2)
plt.title('Actual vs Predicted Load (Validation Set - Last Fold)')
plt.xlabel('Time (15-min intervals)')
plt.ylabel('Load (kW)')
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
plt.figure(figsize=(14, 6))
plt.plot(test_df['Actual'].values, label='Actual Load (Test)', linestyle='-', linewidth=2)
plt.plot(test_df['Predicted'].values, label='Predicted Load (Test)', linestyle='--',
linewidth=2)
plt.title('Actual vs Predicted Load (Test Set)')
plt.xlabel('Time (15-min intervals)')
plt.ylabel('Load (kW)')
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

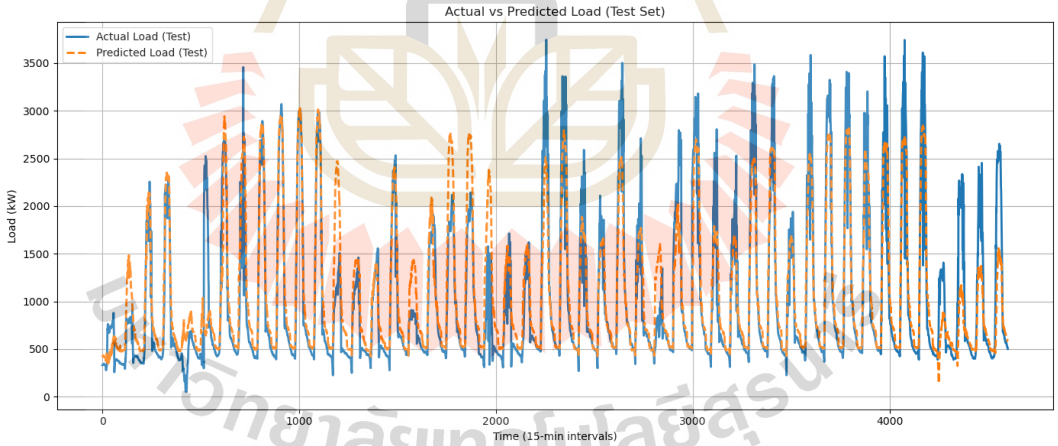
มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX C

Test Set Load Forecasting Performance

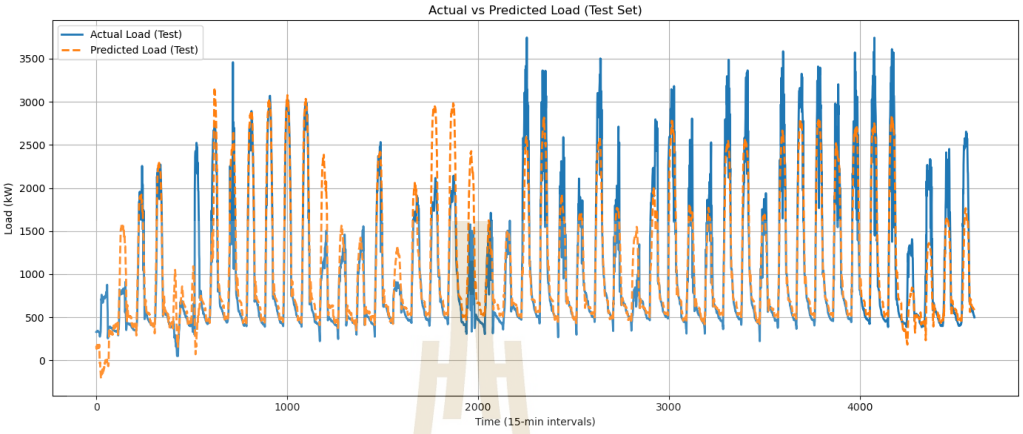


(a)

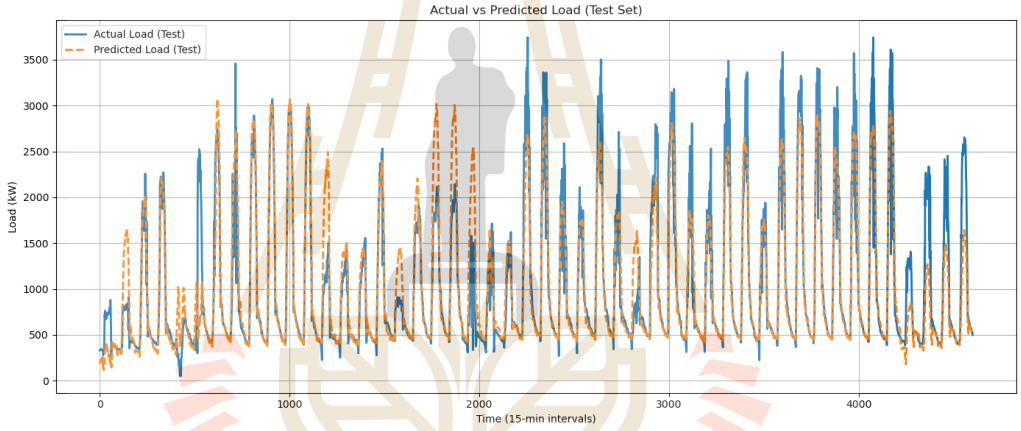


(b)

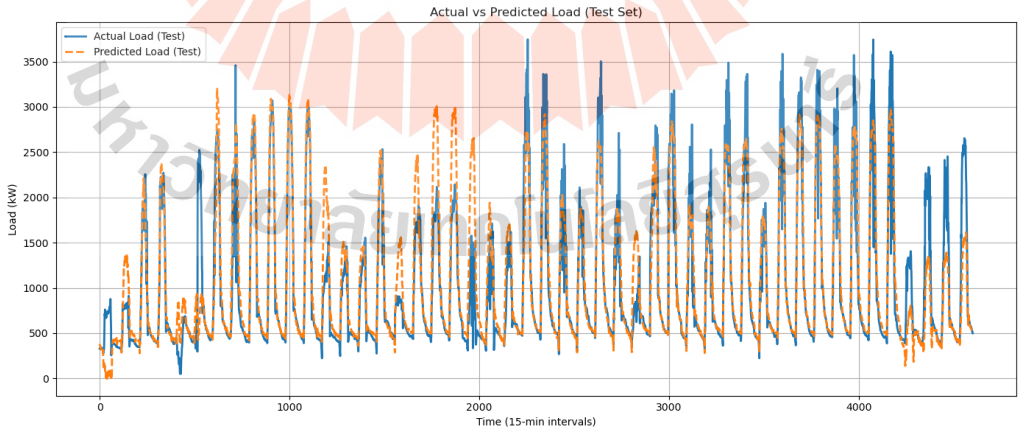
Figure C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM



(c)

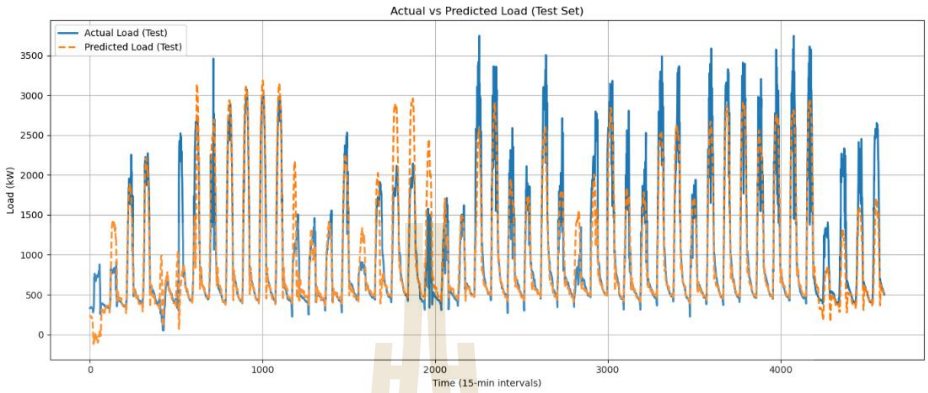


(d)



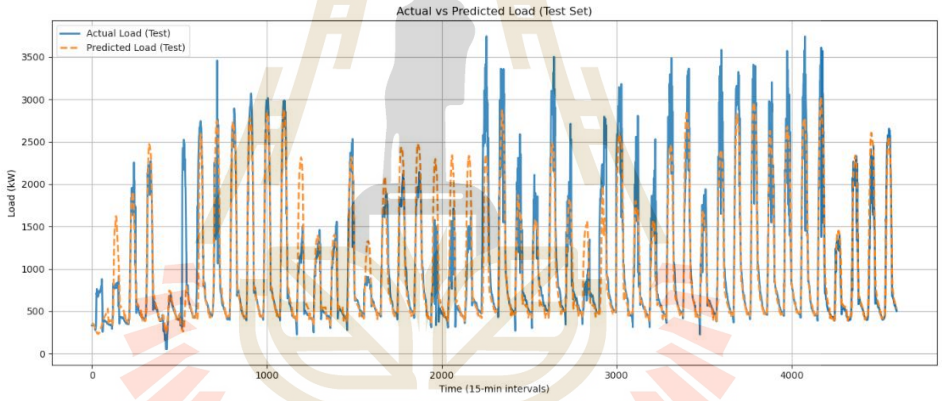
(e)

Figure C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

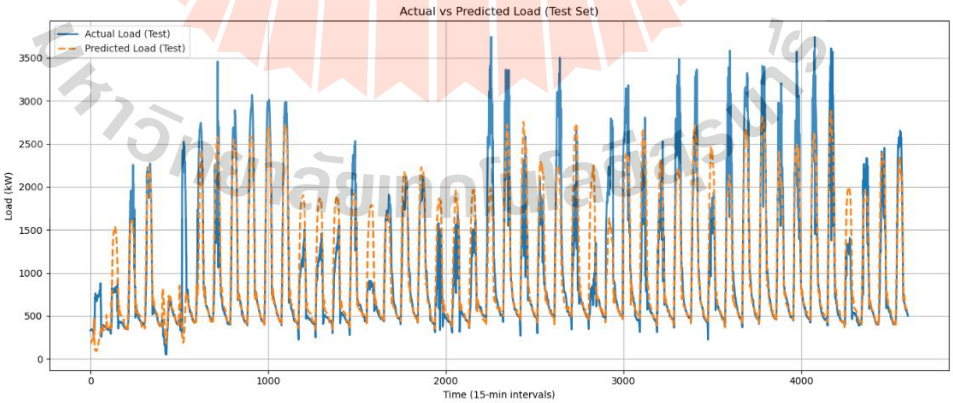


(f)

Figure C.1 Performance evaluation graph of the model's test suite: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

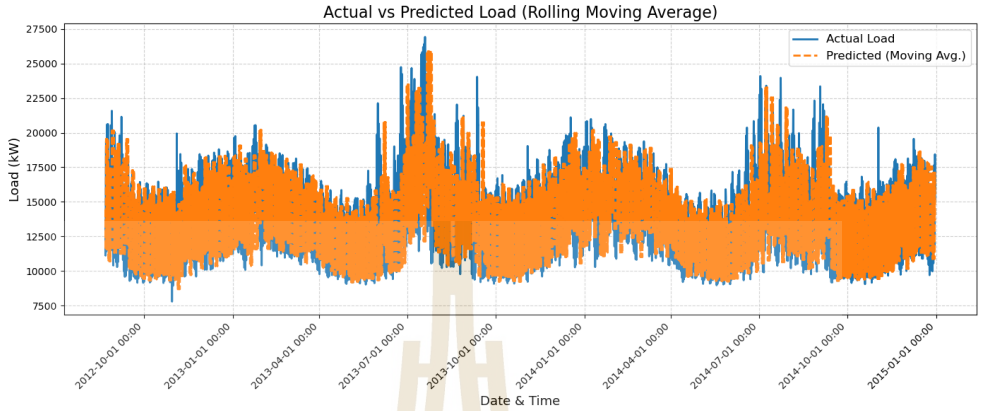


(a)

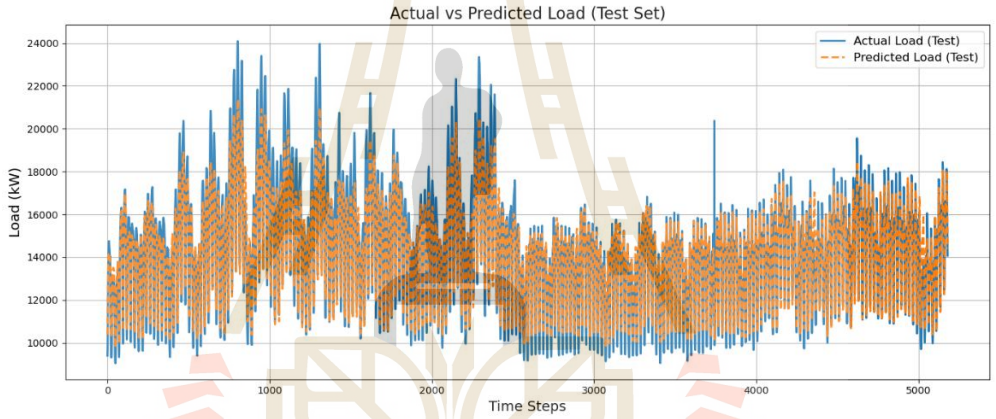


(b)

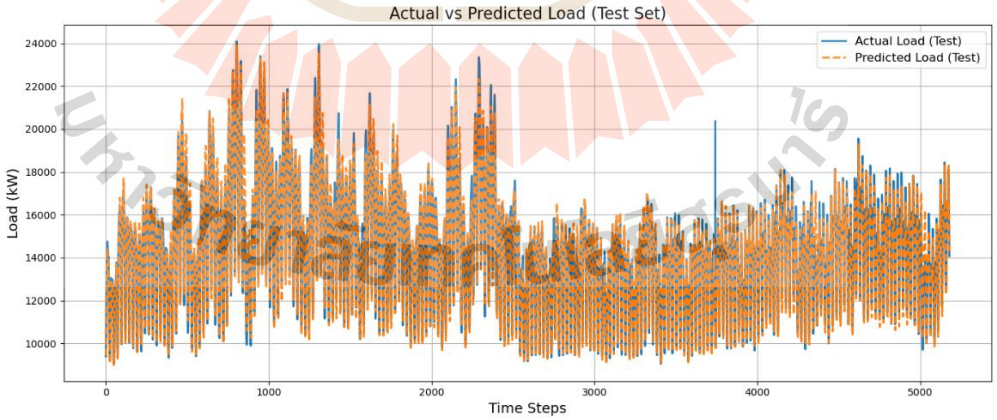
Figure C.2 Performance evaluation graph of the model's test suite: a) Bi-LSTM (Top 3 Features of SHAP) and b) Bi-LSTM (Top 3 Features of MI)



(a)



(b)

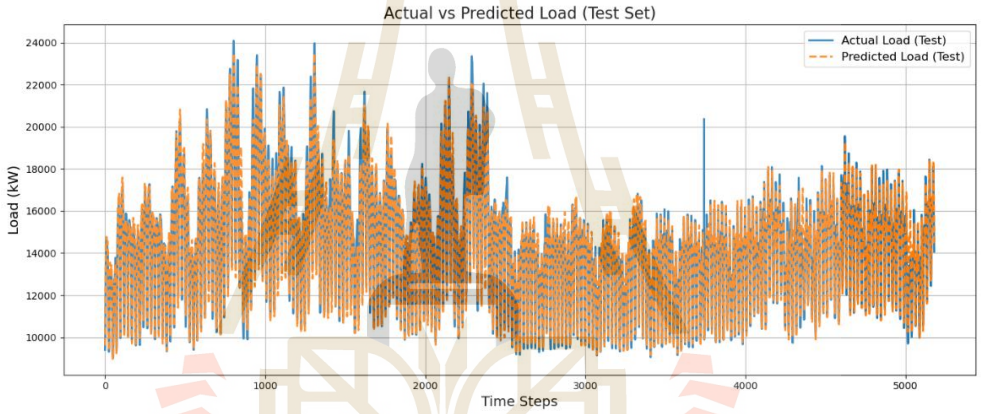


(c)

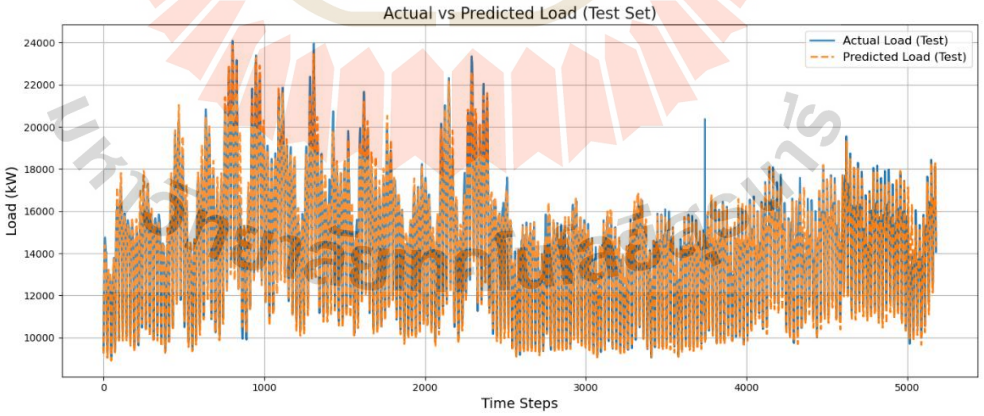
Figure C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features



(d)

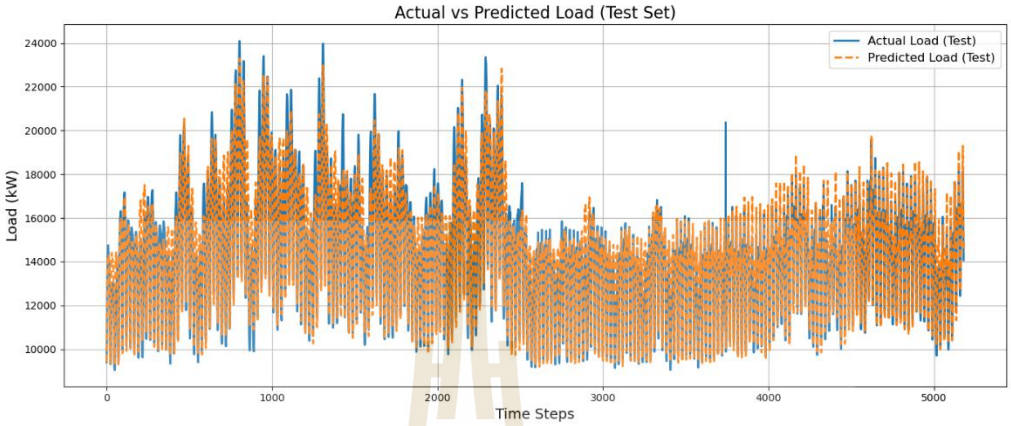


(e)



(f)

Figure C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features (continued)



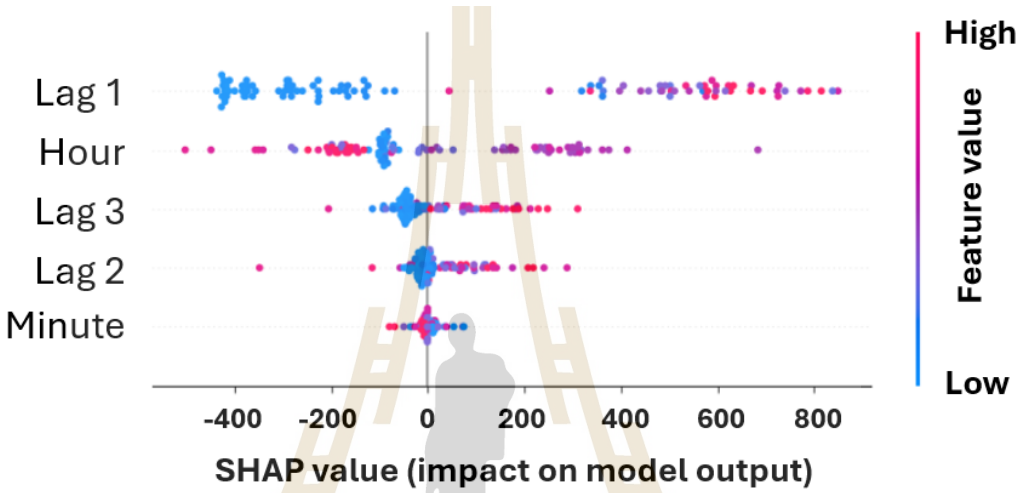
(g)

Figure C.3 Performance evaluation results for test data set of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, f) Attention-LSTM and g) Top 3 Features

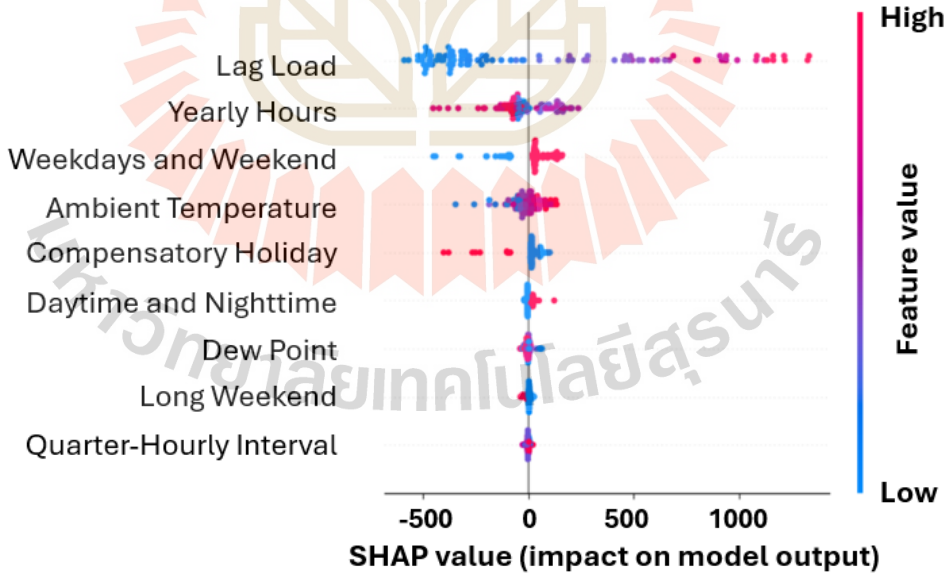
มหาวิทยาลัยเทคโนโลยีสุรนารี

APPENDIX D

SHAP-Based Feature Importance Analysis

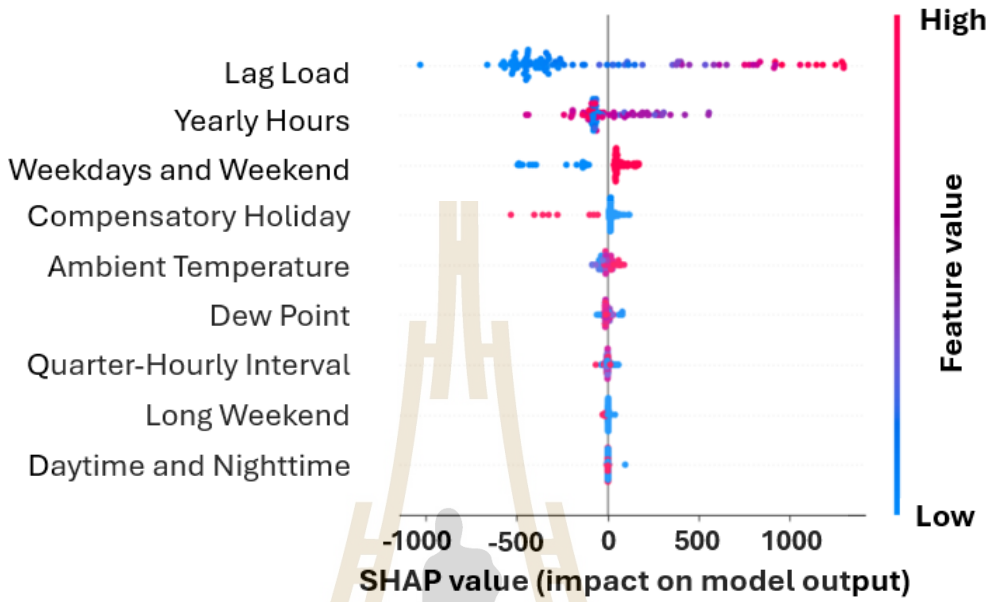


(a)

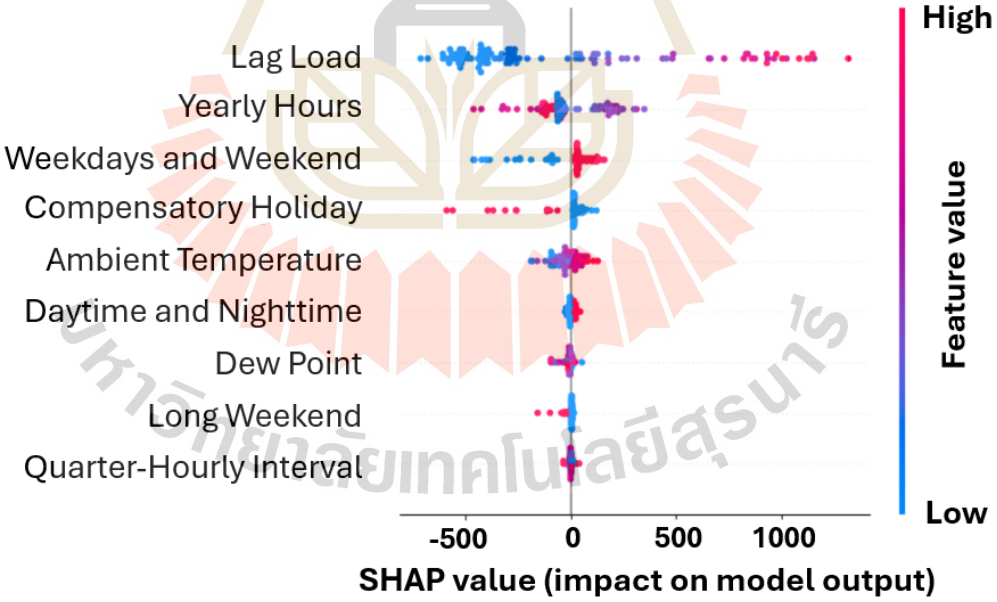


(b)

Figure D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM

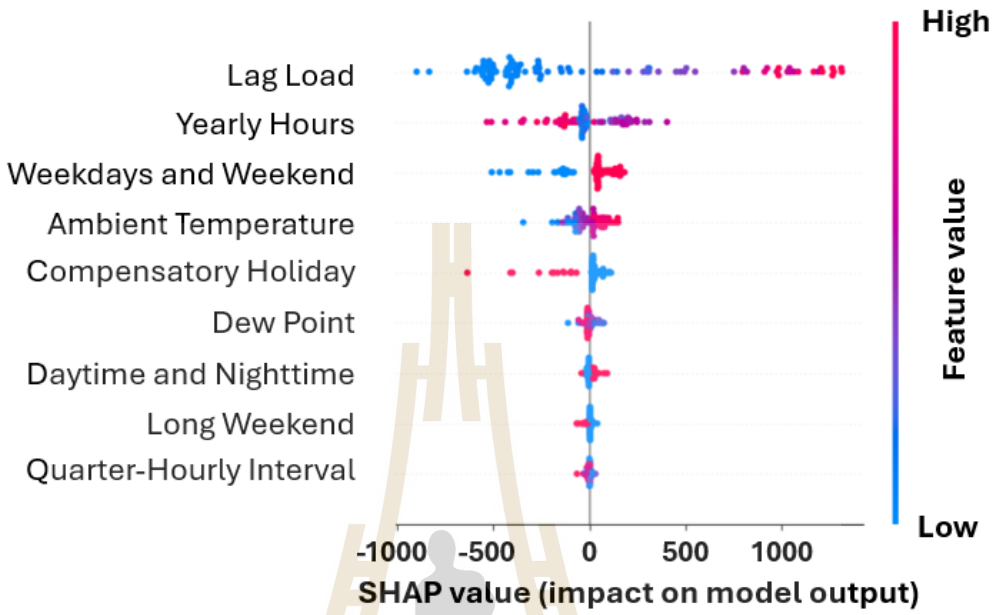


(c)

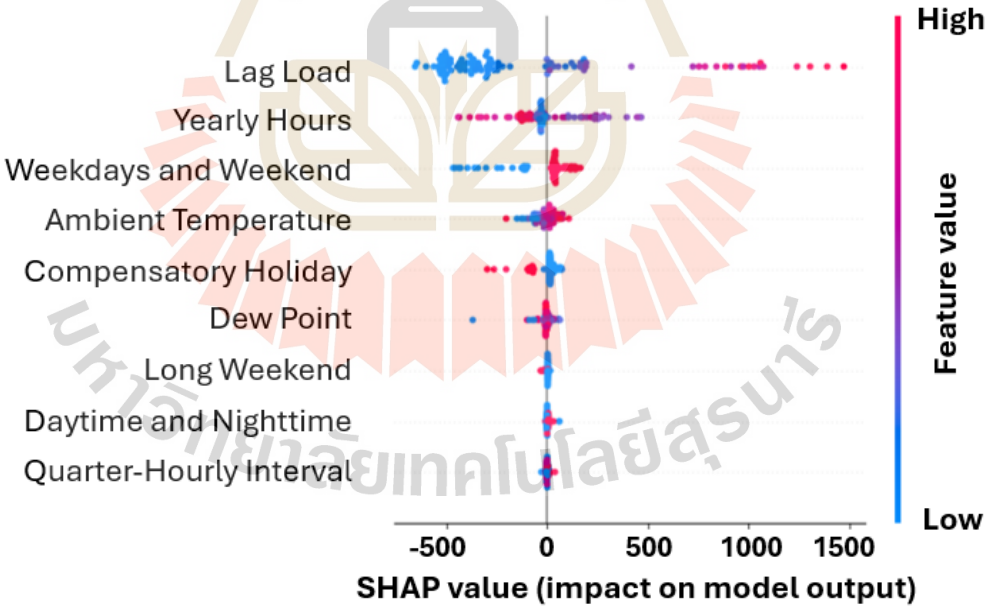


(d)

Figure D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

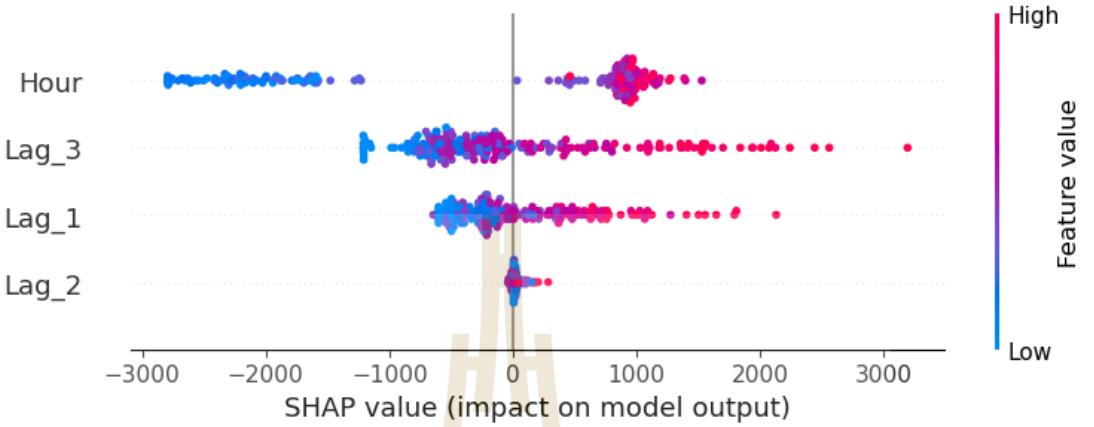


(e)

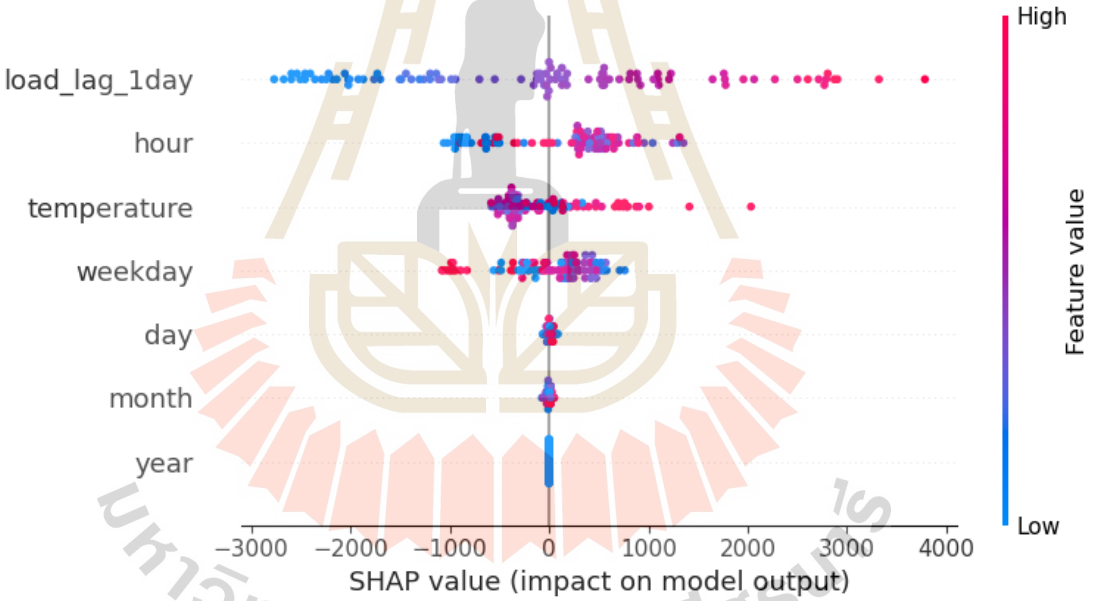


(f)

Figure D.1 SHAP Value of Each Model of SUT's Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

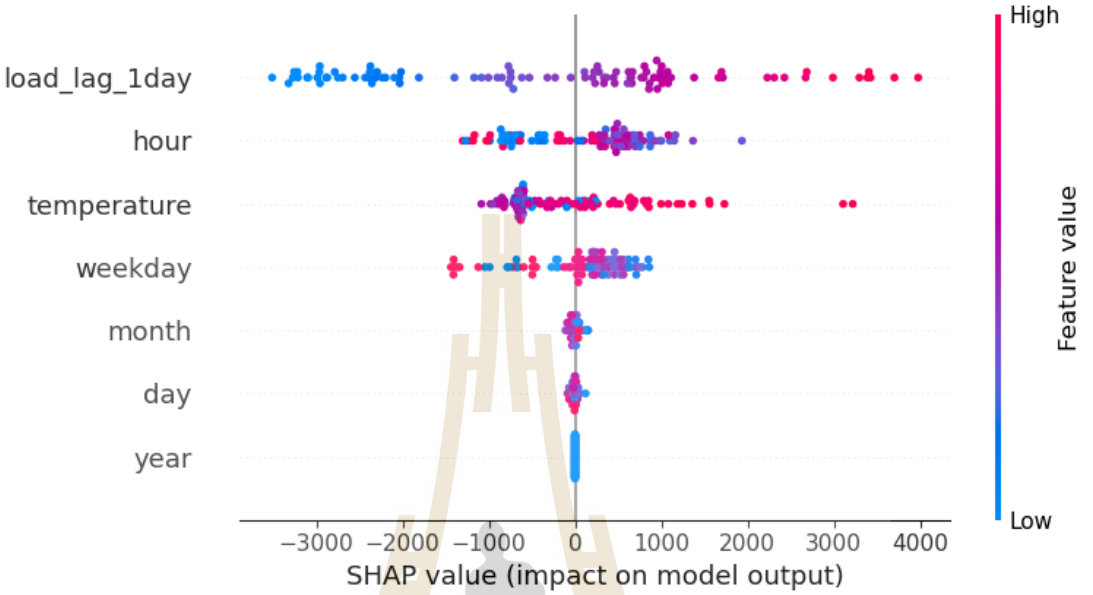


(a)

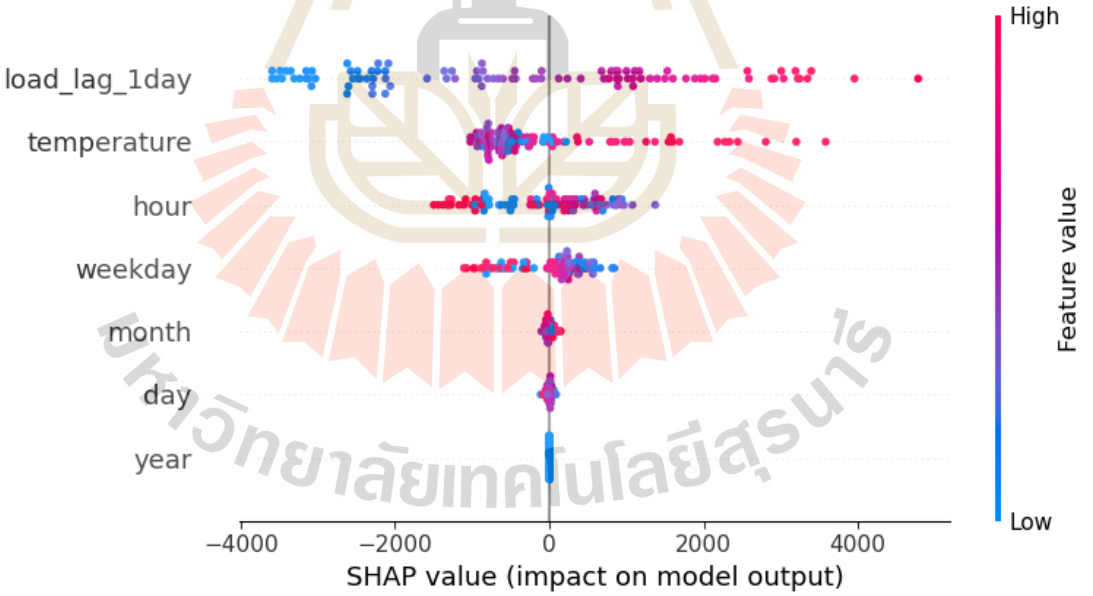


(b)

Figure D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM

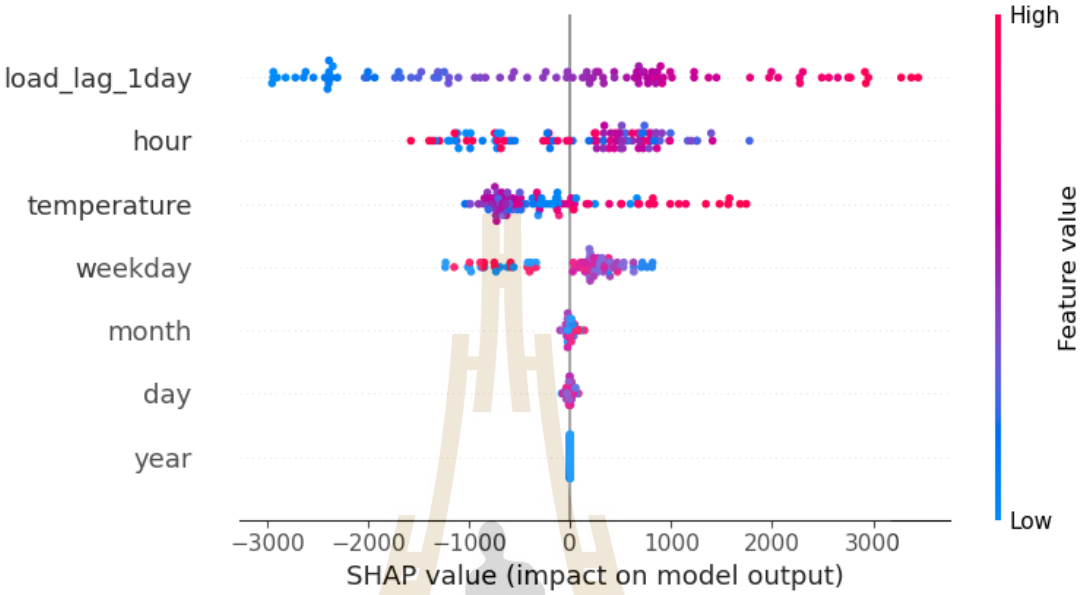


(c)

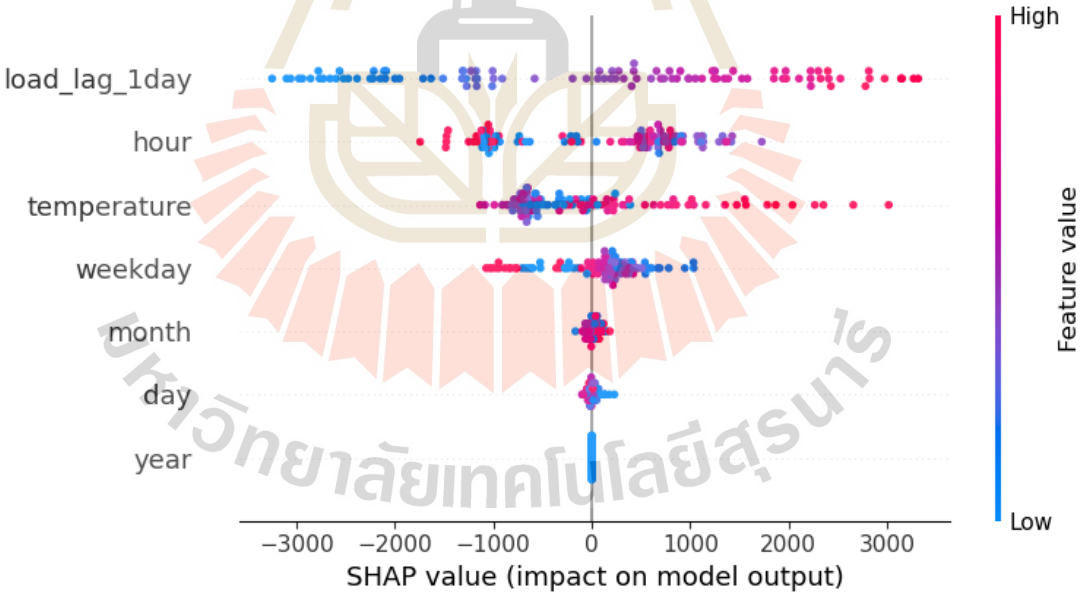


(d)

Figure D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)



(e)



(f)

Figure D.2 SHAP Value of Each Model of ISO-NE Data: a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM (continued)

APPENDIX E

List of publications

N. Thabcha, N. Junhuathon, and K. Chayakulkheeree (2025), Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement: A Case Study of SUT Campus, 2025 13th International Electrical Engineering Congress (iEECON2025), Hua Hin, Thailand.

N. Thabcha, N. Junhuathon, and K. Chayakulkheeree (2025), Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus, ECTI transactions on electrical engineering, electronics, and communications.

มหาวิทยาลัยเทคโนโลยีสุรนารี

Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement: A Case Study of SUT Campus

Nisachon Thabcha
School of electrical engineering
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand
Nisachontabcha@gmail.com

Nitikorn Junhuathon
Department of electrical engineering
Faculty of engineering
Rajamangala University of Technology Thanyaburi
Pathum Thani, Thailand
nitikorn_j@rmutt.ac.th

Keerati Chayakulkheeree*
School of electrical engineering
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand
keerati.ch@sut.ac.th

2025 13th International Electrical Engineering Congress (iEECON) | 979-8-3315-4395-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/iEECON64081.2025.10987599

***Abstract*—This study proposes supervised deep learning (DL) techniques via long short-term memory (LSTM), gate recurrent unit (GRU), Bi-LSTM, and Bi-GRU to forecast electrical loads based on Day-Aware Data Arrangement. The data features are historical electrical loads with arrangement in the form of time series for the model training and evaluation. The proposed model was verified by three accuracy indices: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Square Error (RMSE). Among the study models, the GRU model demonstrates fastest processing speed due to its simpler structure, while LSTM model has the highest forecasting efficiency.**

Keywords—Load forecasting, Long Short-Term Memory, Gated Recurrent Unit, Bidirectional LSTM, Bidirectional GRU

I. Introduction

Efficient energy management, including effective energy planning and load balancing for university or similar facilities, is challenging due to fluctuating or intermittent electrical load data during certain periods. To improve the efficiency of university energy management, accurate short-term load forecasting (STLF) and the development of reliable load forecasting models are essential. However, since traditional forecasting relies on statistical data or overly complex models. Suranaree University of Technology, Nakhon Ratchasima Province, Thailand is an area with continuous and highly fluctuating electricity consumption, both from teaching and various activities held every day. It is necessary to have efficient energy management, reduce unnecessary energy costs through the energy management system (EMS), and help plan and manage energy balance effectively [1]. Short-term load forecasting (STLF) is crucial for EMS, as its accuracy directly impacts system efficiency and performance [2].

Models for load forecasting are crucial, and numerous techniques have been created to achieve this goal. For decades, load forecasting has been a popular topic, and numerous articles using a variety of methods have been published [3], [4]. Regression-based, artificial intelligence, and time-series approaches are the three types of forecasting techniques [5]. ARIMA models and generic statistical analysis are two of the most popular classical time-series techniques. For projecting short-term and medium-term loads, regression techniques are also frequently utilized [6]. These techniques only work with historical load data, are more appropriate for handling load data, and are only applicable to linear data. They might have trouble managing the intricacy of time data, which could result in a large forecasting error for electrical loads [3]. STLF has made extensive use of artificial intelligence and machine learning techniques. These models are capable of handling large and complex datasets and demonstrate adaptability in processing non-linear data. Neural networks (NNs), fuzzy logic systems, and decision trees are a few examples [1]. In electrical load forecasting, NNs are frequently utilized. These are synthetic neural networks designed to resemble the structure of the human brain. The layers that make up their structure are input, concealed, and output layers. They are made to assist in resolving a variety of intricate issues using input data. There are three types of deep learning (DL): supervised, semi-supervised, and unsupervised [7]. A system must learn from the data before it can use supervised deep learning to process a task. The shortcomings of recurrent neural network (RNN) models, particularly with regard to identifying long-term associations in data, are addressed by long short-term memory (LSTM) and gate recurrent unit (GRU) models, which make them better suited for time-sequence data, such as short-term electrical load forecasting with intricate and extremely volatile patterns. Compared to ARIMA, which works well with linear data, they are more adept at handling non-linear and extremely complicated data [8]. Ganguly et al. [2020] presented a comparative analysis of different models for load forecasting using the load profile of a commercial building (amusement park) in San Diego, California, for short-term load forecasting. The results indicate that the LSTM model often yields higher forecast accuracy than the ARIMA and can also be used as part of an EMS. The GRU has a simpler structure than the LSTM model and can be processed quickly, especially for EMS that require real-time response, while the ARIMA may take longer to adjust the model when the data is very volatile [3]. For more accurate forecasting, bidirectional LSTM (Bi-LSTM) and bidirectional GRU (Bi-GRU) models are built from the two models mentioned above. Both of these models perform better than traditional RNN models [9].

Key factors involved in load forecasting, which are considered as load input parameters to improve forecast accuracy and address external influences, include: Temperature, humidity, and weather patterns are important in short-term load forecasting, significantly affecting energy consumption changes. Usage patterns, calendar effects (weekends, public holidays), and social habits are important in capturing energy consumption fluctuations. These input parameters guarantee that the model appropriately adjusts to variations in demand in the actual world [10]. Given the unpredictability and flexibility of renewable energy sources (RESs), it is challenging to estimate consumer demand due to their quick integration into existing power networks. Net forecasting, which uses distinct forecasting procedures for RESs and load, is utilized to get around this problem [7]. By

*Corresponding author

979-8-3315-4395-2/25/$31.00 ©2025 IEEE

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

combining these several forecasts, this method seeks to get the ultimate net forecast. Modern forecasting methods seek to lessen the complexity brought on by external parameter uncertainties by utilizing sophisticated algorithms, real-time load data, and external factors [11].

However, the development of short-term electricity load forecasting models is an important part of the energy management of universities or similar loads, and it is challenging to accurately forecast net electricity demand due to fluctuating or discontinuous electricity load data at certain times. The accuracy of the model is still affected by the characteristics of the data that change over time and days. For example, the energy consumption behavior on weekdays may be different from weekends or public holidays, and it requires a lot of statistical data or complex external factors for traditional forecasting. Therefore, the motivation of this paper is to present a different approach from the traditional method, which is the machine learning or artificial intelligence method using DL through LSTM, GRU, Bi-LSTM, and Bi-GRU models to compare in the same context, making it easy to select the most suitable model and develop the model with the highest accuracy by improving the efficiency of the model through the Day-Aware Data Arrangement approach. It emphasizes that the model can learn the characteristics of the data on each day accurately, reducing the reliance on a large amount of historical load data for forecasting. The accuracy of the electricity forecast results is checked with three accuracy indices: MAE, MAPE, and mean square RMSE, respectively, to solve the above problems.

The highlight of this research is the application of deep learning and Day-Aware Data Arrangement to the university load, which is classified as commercial load, which has fluctuating load usage characteristics because it is a type of load that uses energy related to providing educational services and related activities to create accurate and reliable load forecasting. Short-term electricity forecasting also helps colleges design and improve energy management systems that are more automated and respond to demand more quickly. It also serves as a basis for further research to create smart energy systems in educational institutions and also provides important information for creating appropriate and sustainable energy policies in the future.

This paper is structured into four sections: Section I introduces the study, Section II outlines the methodology, Section III presents the results and discussion, and the final section provides the conclusion.

II. Methodology

A. Data Collection and Preparetion

The datasets collected and used in the forecasting include electric load data from utility meters, which shows historical data from the Automatic Meter Reading (AMR) system maintained by the Provincial Electricity Authority (PEA), which records electricity usage according to load demand at the university, as well as solar energy production data, which records energy production from meters installed in the area with solar panels at Suranaree University of Technology in Thailand. Data regarding compensatory holiday long weekends is gathered from university announcements and the academic calendar. A reliable public website with an API for direct data extraction is the source of historical condition data. Every 15 minutes for a year, from January to December 2023, these statistics are collected and compiled. The electrical load forecasting utilizes 35,040 datasets in total.

In the data cleaning process, which is an important process in preparing various data before using it for forecasting in the model, the following methods are used: Handling missing values (Handling Missing Values) will delete the missing data range or the data range with abnormalities and add new data from the method of calculating the average and transforming the data format to be appropriate (Data Formatting) using Timestamp to help create the Time Factor and Social Factors features, helping the model to learn the relationship of the data better and be more efficient in forecasting.

Since the electrical load in the university is a high PV penetration load with a proportion of solar energy produced and sent to the total power system, the electric load data from utility meters or net load obtained is data that is mixed with solar energy production data or PV generation, making it possible to see electricity usage according to the load demand that is unrealistically low. Therefore, PV Generation is added back to obtain pure load or electrical load data that can be used to assess the actual electricity demand in forecasting, as shown in Equation 1.

$$\text{Actual Load} = \text{Metering Load} + \text{Metering PV Generation} \quad (1)$$

The data is divided by taking the entire cleaned dataset for the year and splitting it into three sets including the training set, which uses 80% of the total data; the validation set, which uses 20% of the data for parameter tuning; and the test set, which consists of a separate portion of data that the model has never seen before, reserved for final evaluation.

B. Feature Extraction

Assigning features to existing data is very important in forecasting electrical load. Because features are variables that will be fed into the model to help the model learn relationships and generate accurate forecast results. Features used in forecasting this type of data are a collection of parameters that are significant to load forecasting and are created by taking into account the context of time and day. Creating features from data can be divided into two types as follows: The first type is creating features that are Day-Aware Data Arrangement, which is a method of arranging data that takes into account the context of time and day to allow the model to learn the characteristics of the data on each day or specific time period. It consists of Hour of Day, which is a value that specifies the hours in one year; Quarter-Hourly Interval, which is a value that specifies the minutes every 15 minutes in each hour; Weekdays and Weekend, which are values used to separate data between weekdays and weekends; Daytime and Nighttime, which are values used to separate data between day and night; Compensatory Holiday, which is a value that specifies whether that day is a holiday or not, Long Weekend, which is a value that specifies that period is a special holiday or festival. The second type is a feature that is weather data, consisting of ambient temperature data and relative humidity data. The number of data sets used to create and divide the feature data sets will be the same as the data used in forecasting electrical load.

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

C. LSTM and Bi-LSTM architecture

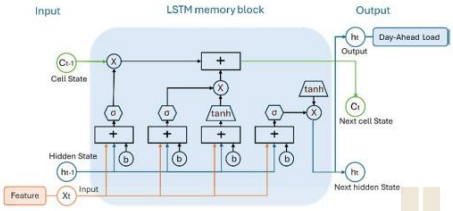


Fig. 1. LSTM architecture.

LSTM is a special kind of RNN that is capable of identifying long-term dependencies. LSTMs are purposefully designed to avoid the problem of long-term reliance. They essentially behave in such a way that their default state is long-term memory retention [12]. For time series forecasting, they are hence the perfect parallel [13]. Fig. 1 describes how LSTM cells are represented graphically.

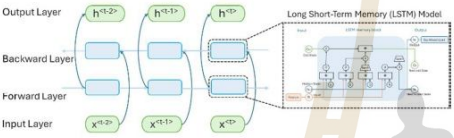


Fig. 2. Bi-LSTM architecture.

Bi-LSTM is a sequence processing model that is made up of two LSTMs, one of which accepts input in a forward direction and the other in a backward manner. Bi-LSTMs give the network access to additional information, which improves the context of the algorithm [14]. The primary advantage is that each input sequence piece includes both current and historical data. Therefore, Bi-LSTM can produce a more pertinent outcome by merging LSTM layers from both directions [15]. The configuration of LSTM layers in Bi-LSTM is shown in Fig. 2.

D. GRU and Bi-GRU architecture

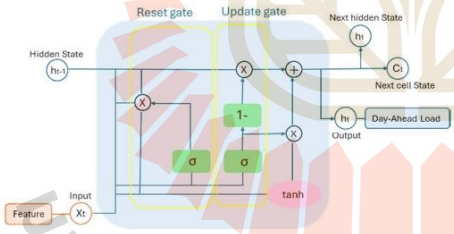


Fig. 3. GRU architecture.

GRU resemble LSTM, which regulates data flow using gates. But with a simpler design, GRU provides an improvement over LSTM. GRU just has a hidden shape, not a distinct cell state like LSTM. The GRU has unrestricted access to all concealed states. Compared to LSTM units, GRUs are easier to train and do not require a memory unit to regulate the data flow [16]. As illustrated in Fig. 3, GRU may train more quickly or employ less data to generalize since it has fewer parameters.

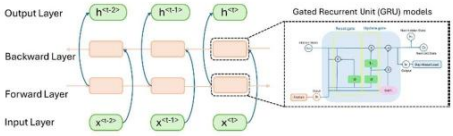


Fig. 4. Bi-GRU architecture.

Bidirectional GRU - A bidirectional GRU, often known as a Bi-GRU, a bidirectional GRU is designed to analyze univariate time series data. This enables the temporal dependencies found in the electrical load data to be accommodated. This integration becomes essential for precise forecasting by utilizing data from both previous and subsequent time periods. As illustrated in Fig. 4, this functionality plays a substantial role in improving forecasting precision [17].

E. Load Forecasting Framework

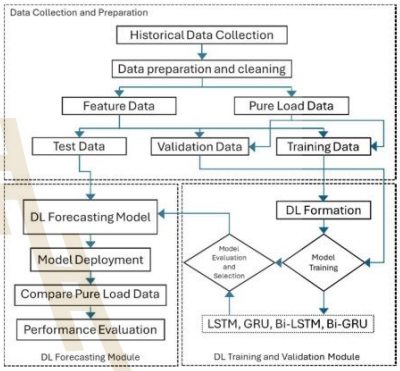


Fig. 5. The Proposed Load Forecasting Framework

The electrical load forecasting, including finding the model with the appropriate accuracy for the university's load data, can be done as follows. Start by collecting all relevant historical data, then clean the data with necessary techniques to facilitate model learning. The data is divided into three groups: the training data group receives feature data as input and pure load data for the model to learn the relationship, then performs the deep learning formation step, which is to design the structure and select the function for the various models and train the model. The models used for training include LSTM, GRU, Bi-LSTM, and Bi-GRU, and they output the load forecast. The validation data group receives the same input as the first group, receiving various models for forecasting. When the output is the load forecast, it will help check the accuracy of model training and prevent overfitting in the model training process. It also helps evaluate the performance of the model during model training of the predicted values and actual load values. Then select the best-performing model as the deep learning forecasting model. The test data group receives features as input, inputs data into the selected model, and outputs daily load forecasts and obtains the developed model. By comparing the predicted values with the actual load values available, the performance of the tested model can be evaluated, as can be seen in Fig. 5.

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

F. Performance Factor

To examine the performance of the proposed forecasting model through evaluation metrics including percentage of MAPE, MAE and RMSE, as shown in Table I.

TABLE I. PERFORMANCE FACTOR

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - y_i')}{y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^{n}\lvert y_i' - y_i\rvert}{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2}$

III. RESULTS AND DISCUSSION

Python 3 software was employed to simulate the Suranaree University of Technology's daily electrical load forecast in order to create a model suitable for the university's electrical loads and evaluate and contrast efficiency using evaluation metrics including MAPE, MAE, and RMSE. The data selected for use was every 15 minutes throughout 2023 and contained, in addition to pure load and feature or electric load data used for forecasting, The models made use of to simulate the results were LSTM, GRU, Bi-LSTM, and Bi-GRU. To get this data ready for entry into the model, it was separated into three sections. This section will show the outcomes of the simulation and different model settings.

TABLE II. MODEL PARAMETER SETTINGS.

Methods	LSTM	GRU	Bi-LSTM	Bi-GRU
Layer	2	2	2	2
Units	100	100	100	100
Dropout rate	0.2	0.2	0.2	0.2
Learning rate	0.001	0.001	0.001	0.001
Batch Size	4	4	4	4
Epochs	50	50	50	50
Optimizer	Adam	Adam	Adam	Adam

TABLE III. EVALUATION RESULTS OF MODEL PERFORMANCE IN VALIDATION METRICS AND MODEL TRAINING TIME.

Methods	Validation Metrics			Model Training
	MAPE (%)	*MAE (kW)*	*RMSE (kW)*	*Time (s)*
LSTM	16.36	227.48	428.84	296.34
GRU	20.61	271.96	464.28	176.76
Bi-LSTM	16.86	241.22	439.58	589.14
Bi-GRU	17.71	252.18	453.94	318.95

A small data set of size four is used to improve the diversity of learning, using 50 epochs to give the model enough time to learn from the training data, and the model structure is set to have 100 units as the number of units in the hidden layer to train different DL models. Two layers of each model are used, a dropout rate of 0.2 is used to prevent overfitting, and Adam Optimizer is used to adjust the model weights, and a learning rate of 0.001 is used as the learning rate for Adam Optimizer, as shown in Table II. The GRU model trained faster than the LSTM model, which took 296.34 seconds. Along with the evaluation of the model's performance in the validation set data, the model training time could indicate the device's usage of resources. The findings of the model's electrical load forecasts will be validated using this data. With MAPE = 16.36%, MAE = 227.49 kW, and RMSE = 428.85 kW, the LSTM model performs best after the performance evaluation. The Bi-LSTM model, with MAPE = 16.86%, MAE = 241.22 kW, and RMSE = 439.5889 kW, is the experimentally efficient model. The two models' training errors are comparable to those of the GRU model (MAPE = 20.61%, MAE = 271.96 kW, and RMSE = 464.29 kW) and Bi-GRU (MAPE = 17.71%, MAE = 252.19 kW, and RMSE = 453.94 kW), according to the three metrics. According to the validation dataset, results demonstrate the extent to which the LSTM and Bi-LSTM models learn and capture relationships. With its speed and predicting accuracy, LSTM offers of potential for real-world use. Table III displays the values in the validation set, which includes the model training settings or the simulation results of the validation metrics' performance.

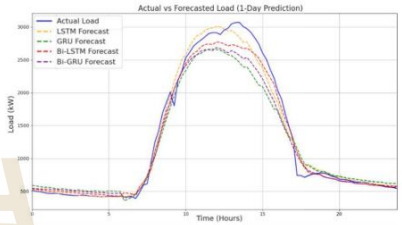


Fig. 6. One-day-ahead electrical load forecast for each model on weekdays

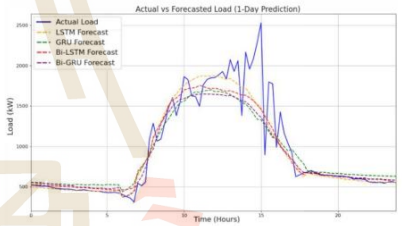


Fig. 7. One-day-ahead electricity load forecast for each model on weekends

After model training and validation, the model is tested in real use. The test set data is divided into two types: weekday load is the load from Monday to Friday, which has a higher total electricity demand than the holiday load. Additionally, weekend load is the load on Saturday or Sunday with the largest variability in electricity demand. The following test days have been selected to evaluate the precision and capacity for forecasting each load model when the electrical load exhibits distinct characteristics: The weekday load is Wednesday, March 1, 2023. And the weekend load is Sunday, October 1, 2023. Forecasting the electrical load output for a day or 24 hours in the future is required, with data being displayed every 15 minutes. There will be a total of 96 data sets. The actual test date is simulated through each model. The results obtained from simulating weekday load are shown in Fig. 6 and the results from simulating holiday load are shown in Fig. 7. The forecast graph will appear. The blue line represents the actual electrical load, or pure load, the yellow line, the green line, the red line, and the purple line. Shows the

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

forecast of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. In the weekdays load forecast, the LSTM model graph gives the forecast that is closest to the actual load graph. The graph is closely related to the actual load graph. The next order has the forecast that deviates from the actual load, which are the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively. However, other models can provide the forecast of the actual load trend. In the part of the weekend's load forecast, the LSTM model graph still gives the forecast that is closest to the actual load graph. However, there are still some load periods that do not give the forecast that is as close to the actual value as it should be because it is a day with a significant load fluctuation. Overall, the LSTM model still performs well. The models whose forecast graphs are closest to the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively.

TABLE IV. PERFORMANCE OF EACH MODEL ON WEEKDAY LOAD

Methods	Model Performance of Weekday Load		
	MAPE (%)	*MAE (kW)*	*RMSE (kW)*
LSTM	6.09	81.51	122.84
GRU	13.29	170.33	242.46
Bi-LSTM	7.38	91.38	131.30
Bi-GRU	9.88	135.16	196.28

TABLE V. PERFORMANCE OF EACH MODEL ON WEEKEND LOAD

Methods	Model Performance of Weekend Load		
	MAPE (%)	*MAE (kW)*	*RMSE (kW)*
LSTM	9.62	115.12	214.53
GRU	15.51	157.18	254.17
Bi-LSTM	11.21	127.00	219.68
Bi-GRU	11.42	138.39	243.42

In addition, the performance of each model was evaluated through all three indicators or through evaluation metrics. The results of the weekday load evaluation are as follows: The LSTM model has a MAPE = 6.09%, MAE = 81.51 kW, and RMSE = 122.84 kW. The GRU model has a MAPE = 13.29%, MAE = 170.33 kW, and RMSE = 242.46 kW. The Bi-LSTM model has a MAPE = 7.38%, MAE = 91.38 kW, and RMSE = 131.30 kW. And the Bi-GRU model has a MAPE = 9.88%, MAE = 135.16 kW, and RMSE = 196.28 kW. Shown in Table IV. The results of the holiday load evaluation are as follows: The LSTM model has a MAPE = 9.62%, MAE = 115.12 kW, and RMSE = 214.53 kW; the GRU model has a MAPE = 15.51%, MAE = 157.18 kW, and RMSE = 254.17 kW; the Bi-LSTM model has a MAPE = 11.21%, MAE = 127.00 kW, and RMSE = 219.68 kW; and the Bi-GRU model has a MAPE = 11.42%, MAE = 138.39 kW, and RMSE = 243.42 kW. Shown in Table V. From the evaluation of the model performance, the results of the weekday load, the LSTM model shows the lowest forecast error when compared to the Bi-LSTM model, Bi-GRU model, and GRU model. For the results of holiday load, the LSTM model still has the lowest error of all metrics when compared to other models. However, it still has a higher error than weekday load because it is a day with high volatility, especially in hours 13 to 18, which has the highest volatility, resulting in the highest forecast error during that period. Overall, the LSTM model still has the best power demand prediction and gives the lowest error for that day. The results show that the most efficient and accurate model for university load data is the LSTM model because the model can learn the patterns and relationships of long-term periods well, including managing the time series data in the data set most efficiently. The weekday load pattern has slightly higher forecast accuracy than the holiday load because the holiday load is more volatile. The next most efficient models are the Bi-LSTM and Bi-GRU models because the type of data that needs to be forecasted may not give enough weight to consider the data from the past and the future, causing the model to learn the relationship and not be able to use its full potential. The last model is the GRU model because it is a simpler model than the LSTM model, which makes it not as good at learning the relationship of the data set, and this type of data may not be suitable for the model. While Suranaree University's power load can be reliably predicted by LSTM and Bi-LSTM models, there are still some restrictions that should be considered when using these models in other situations. To properly understand long-term correlation patterns, the models need a lot of high-quality data. In the case of data-constrained environments, such as small-scale power systems in rural areas or organizations with data shortages, the model performance may be reduced. In addition, Bi-LSTM and Bi-GRU models are complex and require higher computing resources than GRUs, which may not be suitable in systems with limited computing resources, such as IoT devices or edge computing systems. However, the models can be applied in different environments, both in the large-scale energy industry and in distributed energy systems, by adjusting the model structure and parameter settings to suit the data complexity in each context or adding variables such as weather data or sunrise-sunset times to increase the accuracy of predicting fluctuating energy usage. The use of GRU models with higher model training speed in real-time systems may be a suitable option. The results of this study have important implications for the energy sector in many dimensions. The developed electricity load forecasting model can help enhance the efficiency of energy management strategies in various contexts. By precisely predicting energy demand ahead of time, energy suppliers may effectively plan power generation and distribution, minimize overloading, and minimize transmission line losses through integration with smart grid energy management technologies. These models can be used in industrial energy management to predict the energy demand for pieces of equipment or production processes. This helps energy managers plan when to use energy (demand-side management), lower electricity costs during periods of high energy prices, and make the production system more sustainable. In addition, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources such as solar and wind power, which can help reduce energy imbalances and improve the efficiency of renewable energy use, as well as support the reduction of unnecessary energy use as part of the strategy to reduce greenhouse gas emissions in the energy sector.

IV. CONCLUSION

This article presents an alternative approach to developing short-term electric load forecasting models and improving their accuracy through deep learning techniques, including the Day-Aware Data Arrangement method. The most widely used forecasting methods in this field were simulated and compared to evaluate their performance and accuracy. The daily electric load forecasting for Suranaree University of Technology often correlates with past time intervals, allowing LSTM to effectively capture these patterns better than other models. As a result, its load predictions are the closest to the actual values. Adding bidirectional layers to the models did not significantly enhance the results for electric load data. The GRU model,

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

while easier to implement and faster to process, demonstrated lower performance and accuracy compared to the LSTM model for sequential data of this nature. Notwithstanding the encouraging outcomes, this study also discusses significant shortcomings and challenges for further applicability. For maximum performance, high-quality and adequate data are necessary, which might be difficult in contexts with limited resources. Nonetheless, similar models can be used in various energy sectors by reorganizing them, adding new variables, and adjusting the parameter values. The study's findings have important ramifications for the energy industry overall. Planning energy production and distribution effectively, reducing energy losses, and maximizing the integration of renewable energy sources are all made possible by the inclusion of accurate forecasting models, particularly LSTMs, into energy management techniques. Faster GRU models may be suitable for real-time applications where computational resources are limited.

Developing better, scalable forecasting algorithms that adapt to the specific load characteristics of universities and can be applied accurately in other contexts will be a major focus of future research. There is much hope for the design and improvement of energy management systems by combining cutting-edge machine learning techniques with real-time adaptive capabilities. This study provides a basic framework for the development of energy management systems and supporting the growth of smart and sustainable energy infrastructure.

REFERENCES

[1] Ahmad, T., Chen, H., Guo, Y., Wang, J., Shair, J., & Al-Ghamdi, S. G. (2018). Load forecasting for a campus university using ensemble methods based on regression trees. Energies, 11(8), 2038. MDPI. https://doi.org/10.3390/en11082038

[2] T Shepero, M., van der Meer, D., Munkhammar, J., Widegren, K., & Widen, J. (2019, June 23–27). Short-term load forecast for energy management system using neural networks with mutual information method of input selection. 2019 IEEE Milan PowerTech, Milan, Italy, 1–6. IEEE Xplore. https://doi.org/ 10.23919/SICEISCS.2019.8758714

[3] Ganguly, S., Nath, D., & Mandal, K. K. (2020). Bone Fracture Detection and Classification using Deep Learning Approach. 2020 International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control (PARC), Mathura, India, 1–5. IEEE Xplore. https://doi.org/10.1109/PARC49193.2020.236611

[4] Taylor, J. W., & McSharry, P. E. (2007). Short-term load forecasting methods: An evaluation based on European data. IEEE Transactions on Power Systems, 22(4), 2213–2219. DOI: 10.1109/TPWRS.2007.907583

[5] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. PLOS ONE, 13(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

[6] Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. IEEE Transactions on Power Systems, 16(1), 44–55. DOI: 10.1109/59.910780

[7] Vanitha, R., & Jothi Swaroopan, N. M. (2023). Contrast of stacked GRU-RNN and bidirectional LSTM network-based process for forecasting renewable energy for smart grid. 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE). Chennai, India, 1–6. IEEE Xplore. https://doi.org/10.1109/RMKMATE59243.2023.10369285

[8] Doe, J., Smith, A., & Brown, C. (2022). A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing. Energy Systems Research Journal, 18(4), 56–72. https://doi.org/ 10.1109/SAS48726.2020.9220021

[9] Prakash, S., Jalal, A. S., & Pathak, P. (2023, March 3–4). Forecasting COVID-19 pandemic using Prophet, LSTM, hybrid GRU-LSTM, CNN-LSTM, BiLSTM, and stacked-LSTM for India. 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India. https://doi.org/10.1109/ISCON57294.2023.10112065

[10] Ahmad, N., Ghadi, Y., Adnan, M., & Ali, M. (2022). Load forecasting techniques for power system: Research challenges and survey. IEEE Access. Received May 30, 2022, accepted June 28, 2022, published July 1, 2022, current version July 11, 2022. https://doi.org/10.1109/ACCESS.2022.3187839

[11] Patel, R., Sharma, K., Gupta, L., & Singh, P. (2024, January 27–29). A comparative analysis of deep learning-based load forecasting techniques. 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control (PARC), Mathura, India, 1–6. IEEE Xplore. https://doi.org/ 10.1109/AISP61396.2024.10475303

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. https://www.deeplearningbook.org/

[14] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681. https://doi.org/10.1109/78.650093

[15] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5–6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

[16] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv. https://doi.org/10.48550/arXiv.1406.1078

[17] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv. https://doi.org/10.48550/arXiv.1412.3555

Biography



Nisachon Thabcha received her Bachelor of Engineering degree in Electrical Engineering (Second Class Honors) from Suranaree University of Technology, Thailand in 2024. She is currently pursuing her Master's degree in Electrical Engineering at the Faculty of Electrical Engineering, Institute of Engineering, Suranaree University of Technology. Her research interests include deep learing technique application in power system and related fields.



Nitikorn Junhuathon was born in Nakhon Ratchasima in Thailand, on February 23, 1995. He received his B. Eng., M. Eng., and Ph.D. In Electrical Engineering from Suranaree University of Technology (SUT), Thailand, in 2017, 2019 and 2023, respectively. He is currently a lecturer at Department of Electrical Engineering, Faculty of Engineering, Raja Mangala University of Technology Thanyaburi (RMUTT), Thailand. His research interests are machine learning and optimization technique for power forecasting, electric vehicle application, and related fields.



Keerati Chayakulkheeree received his B. Eng. in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL) in 1995, and M. Eng. and D. Eng. degrees in Electric Power System Management from the Asian Institute of Technology (AIT), Thailand, in 1999 and 2004, respectively. He is an Associate Professor at the School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand. His research interests include power system optimization, AI applications in power systems, smart grid technology, power system protection, distributed energy resources, electricity market restructuring, renewable energy integration, and carbon emissions in the power sector.

Authorized licensed use limited to: Suranaree University of Technology provided by UniNet. Downloaded on January 20,2026 at 13:28:34 UTC from IEEE Xplore. Restrictions apply.

Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus

Nisachon Thabcha[1], Nitikorn Junhuathon[2,*], and Keerati Chayakulkheeree[1,*]

[1] School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology
[2] Department of Electrical Engineering, Faculty of engineering, Rajamangala University of Technology Thanyaburi
Email: nisachontabcha@gmail.com, nitikorn_j@rmutt.ac.th, keerati.ch@sut.ac.th
* Corresponding Author

ABSTRACT

This research proposes an approach to develop a one-day-ahead electrical load forecasting model using deep learning techniques. It compares deep learning models with basic models, namely Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM, to improve accuracy and reliability through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis. This approach utilizes data collection and feature engineering to optimize the context of an electrical loads with integrated solar power generation, Using Suranaree University of Technology (SUT) campus electrical system as a case study. Three accuracy indices are examined: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). Among the NN-based models, Bi-LSTM and LSTM tend to provide the best overall forecasting performance from the input data, and the Bi-LSTM model is the most accurate model with the lowest metric value among all models. The lag load feature, or the value of the electrical load in the past one day, is the feature that is most related to the forecasting target and has the most impact on the model's decision.

Keywords: Load Forecasting, Long Short-Term Memory, Bidirectional Long Short-Term Memory, Mutual Information, SHAP, Explainable AI

1. INTRODUCTION

It is well known that electric power is one of the most widely used energy sources and has been used for a long time, both in people's daily lives and well-being and as an important part of national and economic development [1]. The greater the economic growth, the greater the population's demand for electricity [2]. Electric power must, therefore, be reliable and managed with quality [3]. In Thailand, there has been a continuous increase in renewable energy and the introduction of electric vehicles, resulting in unpredictable patterns of electricity usage, with fluctuations that are difficult to manage [4]. Therefore, developing accurate and transparent electricity load forecasting is important for managing electricity to achieve maximum efficiency. It is also linked to the National Energy Plan (NEP2024), which defines five pillars of electric power as follows: Pillar 1: Demand Response & Energy Management System (DR & EMS). Load forecasting is the basic information that the EMS system uses to plan production and load management in advance. This includes coordinating the load response for maximum efficiency, reducing peak demand, and increasing the stability of the power system. Pillar 2: Renewable Energy Forecast (RE Forecast), load forecasting and renewable energy are in tandem in maintaining the power system balance, especially when renewable energy is highly volatile. Accurate load forecasting will reduce the uncertainty in power dispatch and support the adoption of a higher proportion of renewable energy into the system. Pillar 3: Microgrid & Prosumer Local Energy Systems, load forecasting in this context enables the appropriate management of energy resources in each area, reducing the risk of power outages or overloads. Pillar 4: Energy Storage System (ESS) and Pillar 5: EV Integration, although not directly related, load forecasting helps to manage charging and energy storage or discharge behaviour more efficiently. If energy storage systems (batteries) are installed in the future or electric vehicles (EVs) are used at a high level, load forecasting will play a role in designing systems to support charging and energy usage at different times [5].

There are three popular methods for forecasting electricity load. Regression-based methods are considered simple models, such as linear regression, which work well with statistically recorded data and determine the relationship between inputs and outputs. However, complex and non-linear energy demand data used for forecasting can pose problems. Time series methods, typically used with time-dependent data, such as ARIMA models, face challenges when load changes suddenly and due to external factors, making the original data difficult to predict and uncertain [6]. Finally, artificial intelligence (AI) and deep learning (DL) techniques are used. These models are more adaptable to large, voluminous, volatile, and complex data sets. Models belonging to the Neural Network (NN) family are currently trending towards widespread use in forecasting [7]. Table 1 presents a summary of key issues of each research study, including feature engineering, selecting the best model from a comparison of forecasts with other models, metrics used for evaluation, techniques used for feature selection, and

Table 1: *Review the Literature and Summarize the Objectives*

Ref	Best Deep Learning Model	Feature Selection	Model Explainability	RMSE	MAE	MAPE
[4]	LSTM	MI	-	✓	✓	-
[8]	LSTM, Bi-LSTM	-	-	✓	✓	✓
[9]	-	-	SHAP value based	-	-	✓
[10]	RAID	-	SHAP	-	-	✓
[11]	-	Correlation	KAN	✓	-	-
[12]	LSTM-Attention-LSTM	Attention weights	Attention weights	✓	-	✓
[13]	Attention-LSTM	Attention weights	Attention weights	✓	✓	-
[14]	CNN-LSTM	Correlation	-	✓	✓	✓
[15]	Bi-LSTM	MI	-	✓	-	✓
[16]	ANN	MI	-	-	-	✓
[17]	MLP	-	SHAP	✓	-	✓
Proposed Framework	**6 models**	**MI**	**SHAP**	✓	✓	✓

model explainability, to address gaps in research and the objectives of this study.

However, it has been found that many works employ deep learning models in the context of load forecasting. Deep learning models are more effective at learning temporal relationships and non-linear behavior than traditional load forecasting methods, coupled with the selection and design of contextual feature engineering.

However, there is a lack of comparisons between diverse models, including traditional methods and deep learning or NN models, including feedforward models compared to RNNs, models that incorporate RNN models, and hybrid models. Furthermore, there is a gap in predicting features that are effective for nonlinear data, clearly explaining the results of black-box models, and identifying features that truly influence the model's priorities. The trend in load forecasting model development in research focuses on interpretability to enhance both efficiency and transparency. Many studies have focused on using SHAP and other Explainable Artificial Intelligence (XAI) techniques to analyze feature importance, such that the results obtained after model training are designed to directly assess feature importance. However, this lack of integration with methods that filter data before modeling is found. Popular MI methods select accurate features before modeling to reduce model bias. Furthermore, some studies have found that using attention weights can only describe the timesteps that the model focuses on for model explanation. Therefore, our research aims to develop approaches to address these challenges. This leads to the following main objectives:

1) To develop a one-day electricity load forecasting model using deep learning techniques. We will compare deep learning models with basic models, including Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. Specifically, we will examine the model with the highest accuracy based on real-world load data reconstructed from electricity meters and solar panels.

2) To improve accuracy and reliability through the design of appropriate and high-quality features. Mutual Information (MI) is a promising approach for capturing complex nonlinear energy consumption patterns. and systematic evaluation using performance indicators including RMSE, MAE, and MAPE.

3) To improve the transparency and interpretability of the model, SHAP analysis is used to identify influential features and analyze their significance. All features in the dataset can be described to provide a deeper understanding of the model's performance.

This research focuses on the accuracy and reliability of one-day-ahead electricity load forecasting at Suranaree University of Technology (SUT). The project uses real-world load data from the Provincial Electricity Authority (PEA), collected every 15 minutes for a year. Deep learning techniques using artificial neural networks compare models from various categories. Data is prepared, and feature engineering is developed based on the context of the area to reflect the university's actual energy demand. The importance of features can be explained both before and after input into the model for electricity load forecasting. The highlight of this work lies in the systematic design of the forecasting process to achieve both model accuracy and feature importance analysis through the integration of MI and SHAP methods. Most studies use only one technique for feature evaluation and evaluating features with non-NN models due to their inherent black-box nature. However, this work interprets features from deep learning models and conducts two-step feature analysis. The first step, using MI, analyzes the relationship between real data and load, providing a statistical overview. The second step, using SHAP, analyzes the impact on actual model decisions, demonstrating the model's performance on the data, comparing and utilizing the advantages of both methods for optimal forecasting performance. Furthermore, black-box models can be explained and data analyzed before being imported into the model.

At the institutional level, this model provides a more accurate energy planning approach, enabling demand management and cost optimization, as well as more

flexible power system deployment [18]. It also employs a data-driven organization that can support peak demand management [19]. The impact on the energy sector extends beyond university energy management and can be scaled up to other contexts, including smart grid energy management. This approach improves energy management efficiency across various contexts by accurately forecasting energy demand. Energy suppliers can plan energy production and distribution, reduce overload, and effectively reduce energy losses in transmission lines through integration with energy management technologies in smart grids [20]. In the industrial energy management sector, it helps forecast energy demand for equipment or processes, helping energy managers plan when to use energy (demand-side management), reducing electricity costs during periods of high energy prices, and making production systems more sustainable [21]. In the conjunction with the renewable energy sources sector, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources, which can help reduce energy imbalances and improve the efficiency of renewable energy use [22]. In the Greenhouse Gas Emissions sector, it supports reducing unnecessary energy use as part of a strategy to reduce greenhouse gas emissions in the energy sector [23]. This paper is divided into four sections: Section 1 introduces the study and stating the research objectives; Section 2 presents the research methodology used; Section 3 presents the study results and discussion; and Section 4 concludes with a conclusion.

2. METHODOLOGY

2.1 Data Collection and Preparation

Before entering the model for load forecasting, Suranaree University of Technology's electricity consumption data was collected for 1 year, collected every 15 minutes in 2023, totalling 35,040 data sets, which is considered high-resolution data for load forecasting. The data was collected from Automatic Metering (AMR) maintained by the Provincial Electricity Authority (PEA). In addition, solar energy generation data was collected from an Automatic Metering (AMR) maintained by an external company, ENOS, with the same resolution as the electricity load data. Weather data and academic calendar data for that year were collected and prepared to input all this data into the future feature engineering method. The required data is the actual load data. Therefore, the load data obtained from the meter was combined with the solar energy generation data to obtain the data showing the actual load of the university, as shown in Fig. 1.

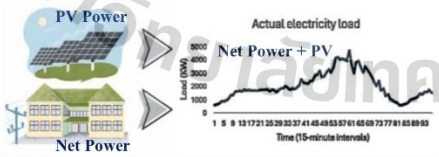


Fig.1: *Preparation of Actual Electricity Load Data*

The preparation of the real electricity load data can be shown on Fig.1. All data used as features and output for the model to learn were divided into two parts: the first part was the data set to be trained, and the other part was partitioned for model validation. Time Series Cross-Validation (TSCV) was performed on this data set, which is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. The training and validation sets were generated using a Time Series Split with 5 folds. The second part was the data set tested on the model after training and validation were completed. It was a data set that the model had never seen before, so the results were unbiased and divided into 4608 data points. For testing, this group of data will be divided into 10 sets; each set will have 3 days. Each set is divided according to the types of data groups, namely long holiday, weekdays during the school term, exam period, summer season, the start of the semester, compensation holiday period, rainy season, weekdays and weekends, term break, and winter season.

2.2 Feature Engineering

In this research, feature engineering has a group of data that has the characteristics of feature creation, which is creating new features from existing time data and converting time data that may not be easy for the model to logical data with contextual meaning. It consists of the following techniques: In the feature transformation method, features that use this method include Weekdays and Weekend, Daytime and Nighttime, Compensatory Holiday, and Long Weekend, which are converting to variable values and binary representation with numbers 0 and 1. In the time-based feature engineering method, features that use this method include yearly hours, which divides the data into meaningful time sequences with 1 year's data every 15 minutes, creating features that allow the model to learn the day and time. The Time Component Extraction method features that use this method include Quarter-Hourly Interval, which extracts time components into separate features, creating new features that describe the sub-patterns that occur in each hour and the lag features method. Features that use this method include Lag Load, which creates features from previous values by taking historical load values to create features. In addition, there is a group of characteristics: external features. External variables affect the forecast of the target variable, such as ambient temperature, an external factor that directly affects energy use behavior, and dew point, an indicator of air humidity.

2.3 Feature Selection and Mutual Information

Mutual Information (MI) captures the relationship of input features to a non-linear data model, shows their importance, and indicates how the feature reduces uncertainty with respect to the forecasting target. It is used as a tool for selecting appropriate forecasting features, especially in tasks where the number of features varies without necessarily being linear. It is calculated using the following Eq.(1).

$$i(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\cdot log\left(\frac{p(x,y)}{p(x)\cdot p(y)}\right), \quad (1)$$

where Y is the target variable, i.e., the future load, X is the input variable, $p(x,y)$ is the joint probability of x and y, and $p(x)$ and $p(y)$ are the marginal probabilities of each variable. The value of MI is expressed as a real number greater than or equal to 0. The higher the value of MI, the more the feature is related to the forecast value. MI can help reduce the size of the data to control both the speed of training and the interpretation of the model. MI is a powerful tool for feature selection for time series models, especially in complex load forecasting tasks that change over time and in different contexts. Selecting appropriate features using MI can make the model more accurate and easier to interpret.

Feature selection in load forecasting each feature plays a physical or behavioural role in load patterns. The Lag Load feature was selected because electrical loads are similar and have daily activities that often repeat within 24 hours, making it a very important representative of daily patterns in forecasting. The Yearly Hours and Quarter-Hourly Interval features help the model understand the time sequence of each day and allow the model to recognize period differences. When used in combination with other features, the model can better learn about daily trends. The Weekdays and Weekend features were selected because electricity usage by electricity consumers differs significantly during holidays and weekdays. The Daytime and Nighttime features were selected because people's behaviour shows higher electricity usage in the morning than at night, so the model should be able to see this difference. The Compensatory Holiday and Long Weekend features have different load patterns during long holidays and special holidays. The Ambient Temperature and Dew Point features are weather-related features that reflect the characteristics of electricity consumers when temperatures and humidity change. They also provide the model with sufficient forecasting capabilities when there is load variation caused by changing weather conditions.

2.4 Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) model is a structure of Recurrent Neural Network (RNN) designed to solve problems and manipulate time-series data. It consists of a Forget Gate that determines which data should be forgotten from the original cell state. Let x_t be the input feature vector at time t, $\sigma(\cdot)$ be the sigmoid activation function, h_{t-1} be the hidden state from time t-1, and b_f be the bias vector of the forget gate, W_f be the learnable weight matrix of the forget gate. The forget-gate output is computed as shown in Eq.(2).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \quad (2)$$

where $f_t \in [0,1]$ is close to 1, indicating retention, while values close to 0 indicate erasure. This helps determine whether data should be retained. The next component, the Input Gate (i_t) and the Candidate Cell State, decides which new data to add to the cell state, as shown in Eq.(3) and Eq.(4).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (3)$$

$$\tilde{C}_t = \sigma(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (4)$$

where i_t controls which new data should be imported, and $\tilde{C}_t$ represents the new data processed (candidate memory). The next component, the Update Cell State, combines the Forget Gate and the Input Gate to create a new cell state, as shown in Eq.(5).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

where the forgotten data is multiplied by f_t, and the new, important data is multiplied by i_t and added together. The final section is the Output Gate (o_t) and Hidden State. These define the data to output as hidden states for future forecasting, as shown in Eq.(6) and Eq.(7).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (6)$$

$$h_t = o_t * tanh(C_t), \quad (7)$$

where h_t is the hidden state, used to predict the electrical load at the next point, as shown in Fig. 2.

2.5 Bidirectional Long Short-Term Memory (Bi-LSTM)

The Bidirectional Long Short-Term Memory (Bi-LSTM) model is a structure that is developed from LSTM to increase the ability to learn sequences of data that have relationships in both the past (past dependencies) and the future (future dependencies) simultaneously. It consists of two LSTM layers that work together: a forward LSTM that processes data from the past to the present and a backward LSTM that processes data from the future back to the present at any time t. The results of both directions are combined into a single hidden state as shown in Eq.(8).

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (8)$$

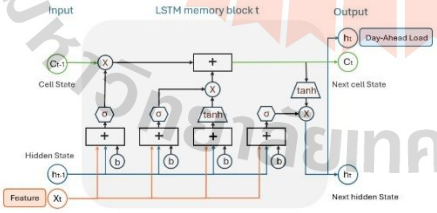


Fig.2: *The LSTM Architecture*

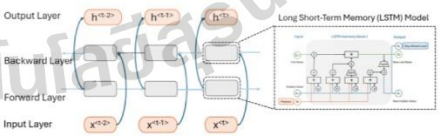


Fig.3: *The Bi-LSTM Architecture*

where $\vec{h}_t$ is the hidden state from the Forward LSTM, $\overleftarrow{h}_t$ is the hidden state from the Backward LSTM, and h_t is the combination of the two results, as shown in Fig. 3.

2.6 Shapley Additive Explanation (SHAP)

SHAP is a method based on the Shapley Value from Cooperative Game Theory, which measures how each feature contributes to the model's decision. SHAP transforms the model description problem into a game in which each feature is a player, and the model's outcome is a reward that must be shared (prediction). The Shapley Value calculation considers all subsets of features (feature coalitions) to assess the impact of each feature. The calculation principle (conceptually) can be described as the Shapley value of feature i is defined by the following Eq.(9).

$$\emptyset i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)], \quad (9)$$

where N is the set of all features, S is the subset of features excluding i, $f(S)$ is the predicted value of the model when using only the feature in S, and $\emptyset i$ is the average of the effects that feature i has on the predicted value when combined with all other features in various orders.

2.7 Load Forecasting Framework

Electrical load forecasting requires the preparation of essential data, including aggregation and collection from reliable sources in the forecasting area, focusing on actual electricity load data from meters. This data is then fed into feature engineering. Two types of data are used: feature generation, which converts raw data into numbers that are easier for the model to understand and allows the model to better learn about data differences, and external feature data, which learns about data trends. This process takes place during the data collection and preparation process. The next step in the simulation process is automatic feature selection using MI to assess the nonlinear relationship between each input variable and the target load. This helps preserve features that are important for forecasting. This reduces data complexity while retaining the most important features. The data is then grouped into three groups for training, validation, and testing, based on proportions. The resulting data is then fed into the model by designing the model, setting various parameters, and training deep learning models in various categories: Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. After training the models on the same dataset for all models, then, in the results and analysis process, the performance is evaluated using standard metrics: root means square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). To enhance the transparency and clarity of the black-box model, the SHAP tool is used to guide decision-making on feature prioritization. This tool identifies the lowest and highest priority features for the final pre-load forecast. Figure 4 shows the entire forecasting process. The first step is the data collection and preparation process. The next step is the simulation process. The last step is the results and analysis process.

2.8 Performance Factor

To examine the performance of the proposed forecasting model through evaluation indicators, namely percentage of MAPE, MAE and RMSE as shown in Table 2, the calculation of the average of each indicator is given by the equation of Average (Avg.), where X is the MAPE, MAE or RMSE indicator on day i and m is the total number of days. Let Y_i denote the actual load at time i, and $\hat{Y}_i$ denote the corresponding predicted load.

Table 2: *Performance Indicators*

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y}_i)}{Y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert}{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$
Average (Avg.)	$\frac{\sum_{i=1}^{m} X_i}{m}$

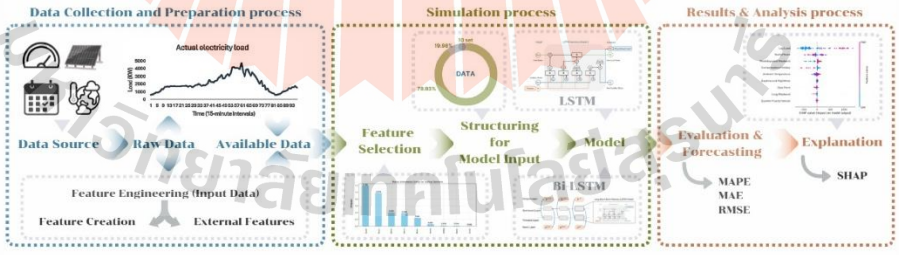


Fig.4: *The Complete Forecasting Process*

3. RESULTS AND DISCUSSION

The electrical load forecast from all the data obtained and the tools mentioned above will be used in simulation using the Python 3 program to show the results comprehensively and explain the accuracy and reliability of the model in the area and context of Suranaree University of Technology. It is a 1-day-ahead load forecast using historical data for 1 year of forecasting in 2023. The details are presented in this section.

The settings for each model are shown in Table 3. All models use similar parameters, namely the Dropout Rate parameter with a value of 0.2 to set the percentage of that unit. To prevent overfitting in the model, the learning rate is set to 0.001 to determine the speed of parameter tuning during learning. The optimizer uses Adam to automatically adjust the learning rate. The batch size, which is the number of data used to calculate the gradient per iteration of parameter tuning, is set to 32. The Epochs, which is the number of iterations to learn the whole data set, is set to 50. The loss function is used to evaluate the model error with MSE. In the different parts, each model is a layer part, which is the basic structure of the model that defines data processing. The DNN model has a Flatten Layer to transform the data and feed it into a Dense Layer, with the number of units in each Dense Layer having the units or the complexity of learning in each layer as 128, 64, and 1. The CNN-LSTM model has Conv1D filters, or the number of filters, as 64. The kernel size is the length of the repeated pattern detection filter with the value of 3. The pool size for reducing the dimensionality is set to 2. The LSTM Units are set to 100 for learning the temporal relationship after Conv1D has extracted features and has a Dense value of 1 in the model part. The Attention-LSTM has an Attention Layer that prioritizes the timesteps that affect the prediction and returns 100 LSTM Units with a Dense of 1. In the LSTM model, the LSTM Unit Capacity is 100, and the Bi-LSTM model has a Bidirectional Wrapper to process two directions and uses 100 LSTM Units in each direction with a Dense of 1.

The selection of the appropriate hyperparameter values for each model was performed using a trial-and-error method. The parameters were adjusted within a specified range and then evaluated by evaluating the minimum MAPE value displayed when each parameter was adjusted within each range. Hyperparameter units were set in a range of 64 to 128, with a best value of 100. The dropout rate was set in a range of 0.1 to 0.3, with the best value of 0.2. The learning rate was set in a range of 0.005 to 0.0005, with a best value of 0.001. Timesteps were set in a range of 4 to 12, with a best value of 8. Batch size was set in a range of 16 to 64, with a best value of 32. Early stopping was set in a range of 5 to 20, with a best value of 10. As shown in Table 4, these hyperparameter sets were then used to simulate the results of each model.

To prioritize the feature data before importing the model with the MI tool, the relationship of the features and the target electrical load dip data groups is measured as both linear and non-linear data. It shows the score ranking of each feature. The Lag Load feature has an MI score of 1.198, the Yearly Hours feature has an MI score of 0.976, the Daytime and Nighttime feature has an MI score of 0.405, the Ambient Temperature feature has an MI score of 0.378, the Dew Point feature has an MI score of 0.241, the Weekdays and Weekend feature has an MI score of 0.055, the Compensatory Holiday feature has an MI score of 0.018, the Long Weekend feature has an MI score of 0.014, and the Quarter-Hourly Interval feature has an MI score of 0.002, as shown in Fig. 5. The introduction of lag load as a feature tends to have a significant relationship between the input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast value. The behaviour of electricity users today and yesterday is similar. The features that are tested with high scores are the Yearly Hours feature, which shows the time position that repeats patterns throughout the year, showing the relationship with the forecast value in a high proportion. These are 2 features that have significantly higher scores than other models. As for other features, the scores are closer to zero and are less relevant to the forecast.

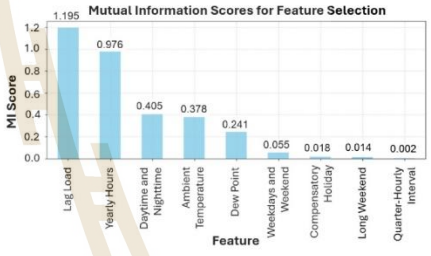


Fig.5: Mutual Information Scores

Table 3: *Parameter Setting*

Model	Parameters Layer
DNN	Flatten, Dense (128), Dense (64), Dense (1), and Units: 128, 64, 1
CNN-LSTM	Conv1D (64), kernel size (3), MaxPooling1D (pool size=2), Dense (1) and LSTM (100)
Attention-LSTM	LSTM (100), return sequences (True), Dense (1) and Attention
LSTM	LSTM (100) and Dense (1)
Bi-LSTM	Bidirectional (LSTM(100)) and Dense(1)

Table 4: Selecting the appropriate hyperparameter

Hyperparameter	Adjustment range	Best parameter values
Units	[64 128]	100
Dropout Rate	[0.1 0.3]	0.2
Learning Rate	[0.0005 0.005]	0.001
Timesteps	[4 12]	8
Batch Size	[16 64]	32
Early Stopping	[5 20]	10

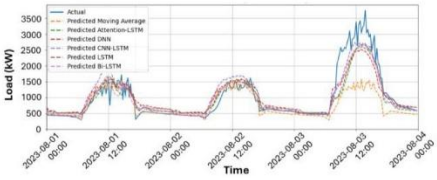


Fig.6: *The Lowest-Performing Forecast Results in the Initial Semester Test Set*

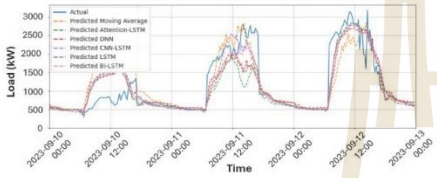


Fig.7: *The Lowest-Performing Forecast Results in the Rainy Season Test Set*

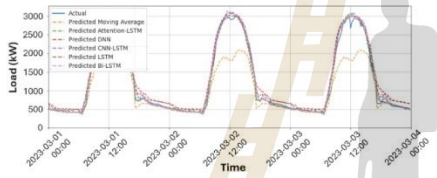


Fig.8: *The Best-Performing Forecasting Results in the Test Set during the Exam Period*

One-day-ahead electrical load forecasting at Suranaree University of Technology can demonstrate the performance of each of the six models through three metrics, compiled into statistical metrics that show the highest, lowest, and average values of each metric from ten test data sets. Important days in the forecast, including the days showing the lowest and highest values of each metric, are also included in the test data set for the model. It is shown in Table 5. The forecasting results show that the moving average forecasting method gives the highest value of all 3 indicators from the tested data set. MAE has a value of 675.72 kW, and RMSE has a value of 945.33 kW on 2023-08-03, which is a date in the set of semesters start dates, as shown in Fig. 6. And MAPE has a value of 78.34% on 2023-09-10, which is a date in the rainy season set, as shown in Fig. 7. The forecasting method with the Bi-LSTM model gives the lowest values of MAE, RMSE, and MAPE indicators from the tested data set, with values of 46.75 kW, 68.35 kW, and 4.82%, respectively, which are a date in the exam set, as shown in Fig. 8. The figures show the load forecast for 1 day in advance for 3 consecutive days. Each model is graphed compared to the actual load graph to show the forecast trend.

Table 5: *Performance of Each Model on 10 Test Sets*

Models	Metric Statistic			
	Metric	Min.	Max.	Avg.
Moving average	MAE (kW)	68.60	675.72	292.63
	RMSE (kW)	87.39	945.33	461.50
	MAPE (%)	5.78	78.34	18.03
Attention-LSTM	MAE (kW)	73.10	385.39	192.96
	RMSE (kW)	114.97	563.92	299.34
	MAPE (%)	5.99	35.40	14.64
DNN	MAE (kW)	97.47	403.80	208.02
	RMSE (kW)	122.82	615.58	308.85
	MAPE (%)	10.16	37.38	17.03
CNN-LSTM	MAE (kW)	68.33	368.25	188.39
	RMSE (kW)	90.55	556.91	293.62
	MAPE (%)	7.21	39.63	14.28
LSTM	MAE (kW)	83.60	362.97	192.74
	RMSE (kW)	105.69	528.20	282.47
	MAPE (%)	9.82	33.84	15.87
Bi-LSTM	MAE (kW)	46.75	373.37	178.63
	RMSE (kW)	68.35	566.79	288.43
	MAPE (%)	4.82	38.74	12.90
Bi-LSTM (Top 3 Features by SHAP)	MAE (kW)	56.95	270.42	183.34
	RMSE (kW)	85.51	388.98	283.21
	MAPE (%)	4.89	36.79	13.99

Since the moving average model relies on the average of historical load patterns, they must be similar at the same time point for accurate forecasting. However, data from the beginning of the semester, the rainy season, or the test data used for forecasting are volatile and unstable. This results in higher forecast error rates than all other models when analyzed for the maximum possible forecast error. The Bi-LSTM model has two layers that process historical data and predict future load trends, enabling it to effectively learn from data patterns that fluctuate with weather and time. Furthermore, the test data during exams or similar data have lag load and yearly hours features that significantly affect the relationship with the forecast at that time, accounting for a significant portion of the MI score. This factor occurs both before and after the forecast point. Therefore, the bidirectional learning method demonstrates the highest forecasting performance compared to all other models when analyzed for the maximum possible model accuracy. The Bi-LSTM model exhibits a lower average error (MAE) value, a percentage error compared to the actual value (MAPE), and a lower proportion of large errors (RMSE) than other models. The model can reduce errors at each time point, reduce forecasts with small errors, and forecast peak load periods more accurately than other models. Therefore, the resulting model is the best-performing model in this context.

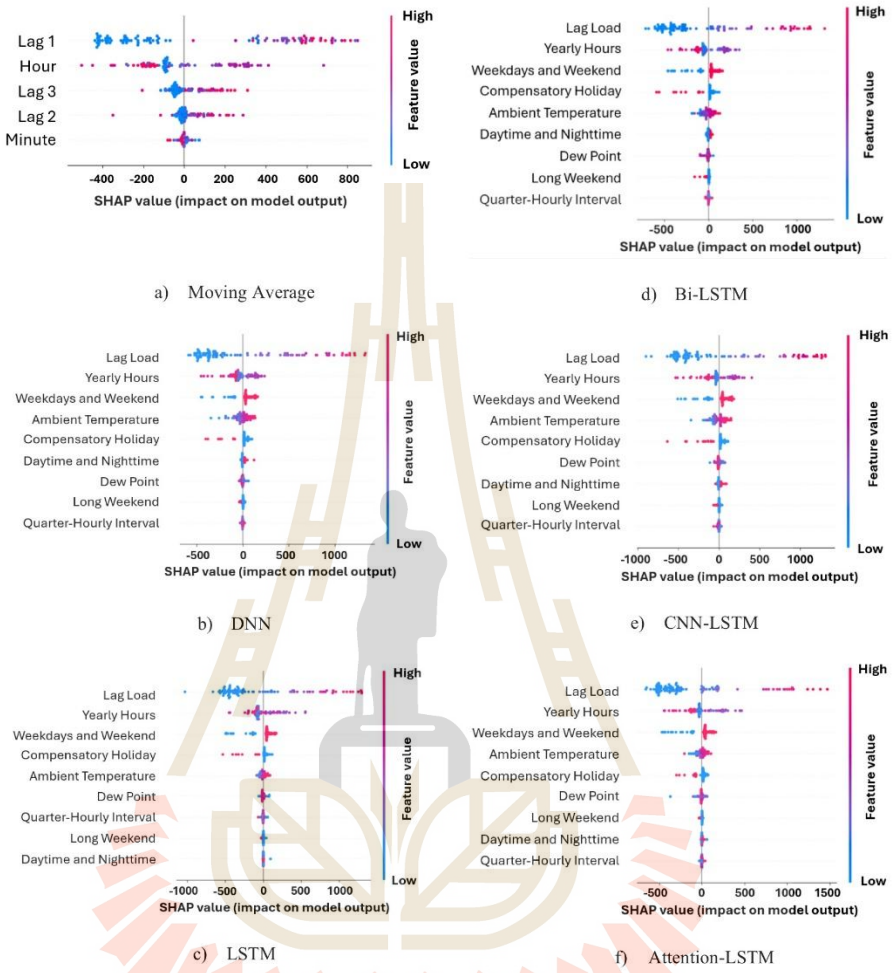


Fig.9: *Shap Value of Each Model:*

a) Moving average, b) DNN, c) LSTM, d) Bi-LSTM, e) CNN-LSTM, and f) Attention-LSTM.

The LSTM model showed the lowest values of all three indicators as follows: MAE was 83.60 kW, RMSE was 105.69 kW, and MAPE was 9.82%. The model has a learning characteristic from the past to the present. There is a layer that remembers various values, which is different from the DNN model in that the features fed into the model are not chronologically ordered, which makes the indicator values higher than the LSTM model. MAE was 97.47 kW, RMSE was 122.82 kW, and MAPE was 10.16%. For the hybrid CNN-LSTM model, CNN was used to capture short-term data patterns, resulting in lower errors. The lowest values of all three indicators were as follows: MAE was 68.33 kW, RMSE was 90.55 kW, and MAPE was 7.21%. For the Attention-LSTM model, developed from a simple LSTM structure, it prioritizes the most effective timestep, showing an MAE of 73.10 kW, an RMSE of 114.97 kW, and an MAPE of 5.99%. Both models exhibit higher forecasting accuracy than the simple LSTM model. The moving average method yields the best indicators, with an MAE of 68.60 kW, an RMSE of 87.39 kW, and an MAPE of 5.78%. These are more effective than some of the models mentioned above. However, when examining

the worst-day recordings, the model achieves higher indicators than all other models. Furthermore, the average test data also has the lowest overall performance.

The average of the indicators from all 10 test data sets, each set having 3 days or 30 data points, can be shown in order of the models with the lowest to highest average MAE indicators as follows: Bi-LSTM, CNN-LSTM, LSTM, Attention-LSTM, DNN, and moving average models with values of 178.63 kW, 188.39 kW, 192.74 kW, 192.96 kW, 208.02 kW, and 292.63 kW, respectively. The order of the models with the lowest to highest average RMSE indicators is as follows: LSTM, Bi-LSTM, CNN-LSTM, Attention-LSTM, DNN, and moving average models with values of 282.47 kW, 288.43 kW, 293.62 kW, 299.34 kW, 308.85 kW, and 461.50 kW, respectively. The model rankings with the lowest to highest average MAPE metrics are shown as Bi-LSTM, CNN-LSTM, Attention-LSTM, LSTM, DNN, and moving average, with values of 12.90%, 14.28%, 14.64%, 15.87%, 17.03%, and 18.03%, respectively, showing the prediction trend of each model from all the input test data. The Bi-LSTM and LSTM models still have good prediction performance overall.

In addition, the Bi-LSTM model was investigated to obtain the most three influential features from SHAP: Lag Load, Yearly Hours, Weekdays, and Weekend. The lowest values of the metrics were obtained as follows: MAE was 56.95 kW, RMSE was 85.51 kW, and MAPE was 4.89%. Compared to the same model set using all 9 features, the results show that the model using only the most important features provides acceptable forecast performance, although its performance is slightly reduced compared to using all features. The Lag Load feature has the highest weight in describing the daily electrical load characteristics. The Yearly Hours, Weekdays, and Weekend features still help the model better understand hourly and weekly patterns. While selecting only 3 important features allows the model to perform well, using all features yields the lowest metric values and the best performance.

The SHAP tool was imported to describe the importance of features that affect the decision of the moving average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models, respectively, as shown in Fig. 9. The x-axis shows the SHAP value, with a positive value indicating a higher prediction value. The y-axis sorts the features by cluster importance from highest to lowest, and the color of the dots indicates the actual feature value, with red having a high value and blue having a low value. The top three features of the NN model with the highest importance are as follows: The Lag Load feature is an important variable that has a great impact on the model's decision because it has the highest SHAP value distribution. There are a lot of red dots to the right, meaning that if yesterday's load is higher, it will also push today's forecast value higher. The Yearly Hours feature is distributed in clusters and has most red dots to the left, meaning that features with high values will lower the forecast value. The Weekdays and Weekend features show that the closer the SHAP value distribution is to zero, the less the feature's impact on the model's decision. In the non-linear features, the SHAP values are widely distributed, with both positive and negative values and a mixture of red and blue dots at the same point. The moving average method is a basic statistical method for forecasting using features Lag 1, Lag 2, and Lag 3 as the electrical load values for the past 1, 2, and 3 days, respectively. The time-related data, namely the Hour and Minute features, are the time data for all 3 days in every hour and minute at the same time of each day. Feature Lag 1 is the feature with the most dispersion and has the most impact on the model's decision.

The importance of ranking of features of NN models by SHAP and MI, where model A is LSTM, B is Bi-LSTM, C is CNN-LSTM, D is Attention-LSTM, E is DNN, and F is MI. From Table 6, number 1 shows the ranking of the most important features, and the higher the ranking number, the importance will decrease until number 9, which has the lowest feature importance to the model. The Bi-LSTM model forecasts have the highest accuracy. The ranking of features from the highest to the lowest is shown as follows: Lag Load, Yearly Hours, Weekdays and Weekend, Compensatory Holiday, Ambient Temperature, Daytime and Nighttime, Dew Point, Long Weekend, and Quarter-Hourly Interval, respectively, were compared for all 6 models. The Bi-LSTM and LSTM models are models that learn the time sequence of data. The 4th feature is Compensatory Holiday, which shows the compensatory holidays of the university reflect the overall electricity consumption behavior. There are red and blue dots clearly separated from the blue ones. The model can recognize clear patterns with this data and associate it with the load drop. This differs from the fifth-ranked feature, ambient temperature. Since alternating points in the same location have a non-linear relationship, fluctuations can make it more difficult for models to capture these patterns.

Table 6: *Feature Importance Ranking of NN Model by SHAP and MI*

Feature	Feature importance ranking					
	A	B	C	D	E	F
Lag Load	1	1	1	1	1	1
Yearly Hours	2	2	2	2	2	2
Weekdays and Weekend	3	3	3	3	3	6
Compensatory Holiday	4	4	5	5	5	7
Ambient Temperature	5	5	4	4	4	4
Daytime and Nighttime	9	6	7	8	6	3
Dew Point	6	7	6	6	7	5
Long Weekend	8	8	8	7	8	8
Quarter-Hourly Interval	7	9	9	9	9	9

A CNN-LSTM model, which captures short-term and long-term patterns well, or an Attention-LSTM model that emphasizes time steps, will better understand this feature. Based on feature priorities, CNN-LSTM and Attention-LSTM models prioritize ambient temperature over compensatory holidays. DNN models can easily learn by finding association weights, do not remember event sequences, and process data without considering time sequences. The SHAP value of the Ambient Temperature

feature, used to describe load changes, is higher than that of the Compensatory Holiday feature, which relies on time sequences and previous behaviors. The Quarter-Hourly Interval feature, also a later feature, is considered a fundamental feature, as it does not directly correlate with load and is a simple feature for the model to learn. From the simulation results, it was found that the MI method gives more importance to the Daytime and Nighttime, Ambient Temperature, and Dew Point features of each model than the SHAP method because these features have more raw data statistics related to the forecasting target, which reflects the actual behavior of electrical loads without learning the model mechanism and not understanding the relationship between the features like SHAP. When using SHAP, the importance is reduced. For Weekdays and Weekend and Compensatory Holiday features, which have lower MI scores, SHAP gives higher importance to both features. This is because the model learns the effect of these features through the more important Lag Load feature, not just learning the features alone. Using SHAP, one can learn features with complex relationships better.

The third feature of the MI method, Daytime and Nighttime, is shown to be highly correlated with load. However, compared to the SHAP method, this feature is ranked low due to the model's inability to extract patterns well. However, the key features, Lag Load and Yearly Hours, ranked first and second in both methods, showed consistent results, indicating that this feature set can definitely improve accuracy and efficiency. Using load data, the MI method screens out potentially important features related to electrical loads, screening relevant and related data before importing into the model and isolating truly irrelevant features. The trend in importance of features is evident, eliminating bias from the model for optimal forecasting performance. However, since it is not yet possible to determine how features affect model decisions, SHAP values are used at each data point in real-world situations to explain the relationship between features and model decisions. This ensures that the model is indeed using those features and provides a deeper explanation of model behavior, demonstrating the importance of features that directly influence the model. Furthermore, under different load conditions, the direction of their impact on forecasting can be identified. Regarding the reliability of black-box models, this can be explained.

4. CONCLUSION

This research proposes the development of a deep learning-based forecasting technique. The accuracy of the model is compared with that of a baseline NN model. The Bi-LSTM model is shown to have the highest accuracy and performance for load forecasting. The data collected through cleaning and feature engineering are then transformed into features ready for input into the forecasting model. The electricity load forecasts are most closely aligned with actual values. Overall, the Bi-LSTM and LSTM models also yield the lowest metric values, averaging across all tested data, indicating the model's potential for forecasting in this context.

Furthermore, the model reliability investigation indicates that transparent feature analysis and model description using MI can explain the relationships between raw data and electricity loads and filter static data for inspection before inputting the model. This reduces the number of features required for model input and reduces training time. SHAP provides in-depth model descriptions, demonstrating the actual use of the model's features, understanding complex features, and increasing model reliability because all features have underlying relationships that can be demonstrated. Using this combined approach, comprehensive coverage is achieved. It provides accurate forecasting results and can also provide causal explanations.

5. ACKNOWLEDGEMENT

This work is fully supported by Suranaree University of Technology.

6. REFERENCES

[1] International Energy Agency (IEA), World Energy Outlook 2023. Paris, France: IEA, 2023. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2023

[2] U.S. Energy Information Administration, *Electricity explained: Factors affecting electricity consumption*, 2023. [Online]. Available: https://www.eia.gov/energyexplained/electricity/

[3] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2019.

[4] T. Panapongpakorn and D. Banjerdpongchai, "Short-term load forecast for energy management system using neural networks with mutual information method of input selection," in *Proc. SICE International Symposium on Control Systems (SICE ISCS)*, Tokyo, Japan, 2019, pp. 13–18.

[5] Ministry of Energy Thailand, *Thailand Smart Grid Development Plan (2022–2031)*, Bangkok, Thailand, 2022. [Online]. Available: https://www.energy.go.th

[6] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, 2001.

[7] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, Mar. 2018.

[8] S. Atef and A. B. Eltawil, "Assessment of stacked unidirectional and bidirectional long short-term memory networks for electricity load forecasting," *Electrical Power Systems Research*, vol. 189, Dec. 2020.

[9] Y.-G. Lee, J.-Y. Oh, D. Kim, and G. Kim, "SHAP value-based feature importance analysis for short-term

load forecasting," *Applied Science and Convergence Technology*, vol. 18, pp. 579–588, 2023.

[10] J. Jang, W. Jeong, S. Kim, B. Lee, M. Lee, and J. Moon, "RAID: Robust and interpretable daily peak load forecasting via multiple deep neural networks and Shapley values," *Sustainability*, vol. 15, no. 8, 2023.

[11] X. Liu, Z. Yang, Y. Guo, Z. Li, and X. Xu, "A novel correlation features self-assigned Kolmogorov-Arnold Networks for multi-energy load forecasting in integrated energy systems," *Energy Conversion and Management*, vol. 300, 2024.

[12] X. Wen and W. Li, "Time Series Prediction Based on LSTM-Attention-LSTM Model," *IEEE Access*, vol. 11, pp. 48322–48331, 2023.

[13] P. Wu, Z. Huang, Y. Pian, L. Xu, J. Li, and K. Chen, "A Combined Deep Learning Method with Attention-Based LSTM Model for Short-Term Traffic Speed Forecasting," *Journal of Advanced Transportation*, vol. 2020, pp. 1–12, 2020.

[14] K. Ullah et al., "Short-Term Load Forecasting: A Comprehensive Review and Simulation Study With CNN-LSTM Hybrids Approach," *IEEE Access*, vol. 12, pp. 111858–111881, 2024.

[15] S. Hu et al., "Short-Term Load Forecasting Based on Mutual Information and BI-LSTM Considering Fluctuation in Importance Values of Features," *IEEE Access*, vol. 12, pp. 23653–23665, 2024.

[16] R. Azimi, A. Amjady, and A. Daraeepour, "Mutual Information-Based Inputs Selection for Electric Load Time Series Forecasting," *Entropy*, vol. 15, no. 3, pp. 926–942, Mar. 2013.

[17] D. A. Bolstad, U. Cali, M. Kuzlu, and U. Halden, "Day-ahead Load Forecasting using Explainable Artificial Intelligence," in *Proc. IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, New Orleans, LA, USA, 2022, pp. 1–5.

[18] M. Ramya and T. Devaraju, "Dynamic scheduling and islanding energy management in microgrids using deep learning and adaptive reinforcement learning techniques," in *Proc. IEEE International Conference on Intelligent Systems and Sustainable Innovation (ICISS)*, 2025.

[19] P. Shanmugapriya, D. R. P. R., S. V. S., and M. SureshKumar, "Effective approach for energy management system using machine learning algorithm," in *Proc. IEEE International Conference on Power, Energy, Control and Transportation Systems (ICPECTS)*, 2024.

[20] J. Zhang, S. Shi, and N. Zhang, "Mitigation strategies for greenhouse gas emissions and energy consumption based on deep learning," in *Proc. IEEE International Conference on New Perspectives in Smart Industrial Future (NPSIF)*, 2024.

[21] J. D. Joy, M. R. Hossain and A. Chowdhury, "A Hybrid Deep Learning Model for Day Ahead Short-Term Load Forecasting," 2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON), Rajshahi, Bangladesh, 2024, pp. 1-6.

[22] M. A. Hassan, A. A. Eladl, B. E. Sedhom, and M. A. Saeed, "A new optimal centralized demand side management for a campus smart microgrid," in *Proc. IEEE International Symposium on Industrial Electronics (ISIE)*, 2023.

[23] S. Akhtar, M. Z. bin Sujod, and S. S. H. Rizvi, "A novel deep learning architecture for data-driven energy efficiency management (D2EEM) – systematic survey," in *Proc. IEEE International Conference on Engineering and Emerging Technologies (ICEET)*, 2021.

Biography



Nisachon Thabcha received her Bachelor of Engineering degree in Electrical Engineering (Second Class Honors) from Suranaree University of Technology, Thailand in 2024. She is currently pursuing her Master's degree in Electrical Engineering at the Faculty of Electrical Engineering, Institute of Engineering, Suranaree University of Technology. Her current research interests focus on power systems and the application of artificial intelligence in power systems.



Nitikorn Junhuathon was born in Nakhon Ratchasima in Thailand, on February 23, 1995. He received his B. Eng., M. Eng., and Ph.D. In Electrical Engineering from Suranaree University of Technology (SUT), Thailand, in 2017, 2019 and 2023, respectively. He is currently a lecturer at Department of Electrical Engineering, Faculty of Engineering, Raja Mangala University of Technology Thanyaburi (RMUTT), Thailand. His research interests are machine learning and optimization technique for power forecasting, electric vehicle application, and related fields.



Keerati Chayakulkheeree received his B. Eng. in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL) in 1995, and M. Eng. and D. Eng. degrees in Electric Power System Management from the Asian Institute of Technology (AIT), Thailand, in 1999 and 2004, respectively. He is an Associate Professor at the School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand. His research interests include power system optimization, AI applications in power systems, smart grid technology, power system protection, distributed energy resources, electricity market restructuring, renewable energy integration, and carbon emissions in the power sector.

BIOGRAPHY

Miss Nisachon Thabcha received her Bachelor of Engineering degree in Electrical Engineering (Second Class Honors) from Suranaree University of Technology, Thailand in 2024. She is currently pursuing her master's degree in electrical engineering at the Faculty of Electrical Engineering, Institute of Engineering, Suranaree University of Technology. During her graduate studies, she conducted research on short-term electricity load forecasting using supervised deep learning techniques, with a particular focus on time-series modeling, feature engineering, and model interpretability. Her research involved the application of advanced recurrent neural network architectures, including CNN-LSTM and Bi-LSTM, to high-resolution load data combined with temporal and weather-related features.

In addition to her research activities, she served as a teaching assistant for courses related to electrical engineering and power system fundamentals, which enhanced her skills in academic coordination, technical communication, and instructional support. She has presented her research findings at international academic conferences, including the International Electrical Engineering Congress (iEECON), gaining valuable experience in professional scholarly communication.

Her research interests include electricity load forecasting, smart grid applications, data-driven energy management, and interpretable artificial intelligence for power systems. She is passionate about contributing to the advancement of smart grid technology and sustainable energy infrastructure in Thailand and aspires to continue working in this field to support national power system planning and distribution development goals.