ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS FOR AGRICULTURE MONITORING

Supakit Siripanadorn

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Telecommunication Engineering
Suranaree University of Technology
Academic Year 2010

การตรวจจับความผิดปกติในโครงข่ายตัวตรวจรู้ไร้สาย เพื่อการเฝ้าระวังเชิงการเกษตร

นายศุภกิตติ์ ศิริพนาดร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2553

ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS FOR AGRICULTURE MONITORING

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

Thesis Examining Committee

(Asst. Prof. Dr. Peerapong Uthansakul)
Chairperson

(Asst. Prof. Dr. Wipawee Hattagam)
Member (Thesis Advisor)

(Assoc. Prof. Dr. Neung Teaumroong)
Member

(Dr. Somsak Vanit-Anunchai)
Member

(Dr. Wut Dankittikul)
Acting Vice Rector for Academic Affairs

(Assoc. Prof. Dr. Vorapot Khompis)
Dean of Institute of Engineering

ศุภกิตติ์ ศิริพนาดร : การตรวจจับความผิดปกติในโครงข่ายตัวตรวจรู้ไร้สายเพื่อการ
เฝ้าระวังเชิงการเกษตร (ANOMALY DETECTION IN WIRELESS SENSOR
NETWORKS FOR AGRICULTURE MONITORING) อาจารย์ที่ปรึกษา :
ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม, 98 หน้า.

โครงข่ายตัวตรวจรู้ไร้สายได้ถูกพัฒนา และนำมาใช้งานเพื่อการเฝ้าระวังเชิงการเกษตรอย่างกว้างขวาง โดยโครงข่ายตัวตรวจรู้ไร้สายมีความสามารถในการเฝ้าระวัง และเก็บข้อมูลทางกายภาพภายในพื้นที่เฉพาะ หรือสิ่งแวดล้อมที่สนใจ ดังนั้นโครงข่ายตัวตรวจรู้ไร้สายจึงเปรียบเสมือนฐานข้อมูลขนาดใหญ่ ซึ่งข้อมูลที่เก็บมานั้นอาจจะเกิดความผิดปกติอันเนื่องมาจากความผิดปกติของตัวตรวจรู้ หรือปรากฏการณ์ทางธรรมชาติที่ผิดปกติ หากส่งข้อมูลทั้งหมดซึ่งมีปริมาณมาก จะทำให้สูญเสียพลังงานเป็นอย่างมาก ดังนั้นในการที่จะลดปริมาณการใช้พลังงานในการส่งข้อมูล ควรมีการจัดการข้อมูลก่อนการส่งข้อมูลดังกล่าว โดยยังคงความแม่นยำในการตรวจจับความผิดปกติ

วัตถุประสงค์ของงานวิจัย คือการนำเสนอกระบวนการการตรวจจับความผิดปกติที่แม่นยำในขณะที่สามารถลดการใช้พลังงานในการส่งข้อมูล ณ สถานีฐาน งานวิจัยนี้ได้นำเสนอกระบวนการตรวจจับความผิดปกติโดยใช้ Self-Organizing Map (SOM) และ Discrete Wavelet Transform (DWT) ในการลดขนาดของข้อมูลก่อนทำการส่ง ซึ่งข้อมูลดังกล่าวได้มาจากการสังเคราะห์ และจากเครือข่ายตัวตรวจรู้ไร้สายในสภาพแวดล้อมจริง

การทดลองแบ่งออกเป็น 3 การทดลอง ประกอบด้วย การทดลองที่ 1 ซึ่งแทรกความผิดปกติที่จำลองขึ้น เข้าไปในข้อมูลจำลอง และข้อมูลจากสิ่งแวดล้อมจริง ซึ่งกระบวนการที่นำได้เสนอ (SOMDWT) สามารถตรวจจับความผิดปกติที่เกิดขึ้นจริงได้ 65% และ 69% ในกรณีของข้อมูลสังเคราะห์ที่ถูกแทรกด้วยความผิดปกติแบบสปาร์ส (Sparse faults) และแบบเบิรสต์ (Bursty faults) และ 67% และ 80% สำหรับข้อมูลจากสิ่งแวดล้อมจริง ที่ถูกแทรกความผิดปกติแบบสปาร์ส และแบบเบิรสต์ ตามลำดับ การทดลองที่ 2 ซึ่งตรวจจับความผิดปกติจากข้อมูลจริงที่ได้มาจากโครงข่ายตัวตรวจรู้ไร้สายในสถานที่ต่าง ๆ กัน ประกอบด้วย NAMOS INTEL SensorScope weather station no.39 และ SensorScope pdg-2008 จากผลการทดลองพบว่ากระบวนการตรวจจับความผิดปกติที่ได้นำเสนอสามารถตรวจจับความผิดปกติที่เกิดขึ้นจริงได้ 99% สำหรับชุดข้อมูล NAMOS 100% สำหรับชุดข้อมูล INTEL 83% สำหรับชุดข้อมูล SensorScope weather station no.39 และ 100% สำหรับชุดข้อมูล SensorScope pdg-2008 ตามลำดับ การทดลองสุดท้ายคือ การตรวจจับความผิดปกติจากชุดข้อมูลที่บันทึกมาจากโรงงานต้นแบบผลิตปุ๋ยอินทรีย์ชีวภาพที่ฟาร์มมหาวิทยาลัยเทคโนโลยีสุรนารี โดยผลการตรวจจับความผิดปกติที่ได้จาก

กระบวนการที่นำเสนอมีประสิทธิภาพใกล้เคียงกับ SOM และมีประสิทธิภาพสูงกว่า DWT ถึง 75% จากผลการทดลองพบว่า กระบวนการที่ได้นำเสนอสามารถคงประสิทธิภาพในการตรวจจับความผิดปกติในขณะที่ใช้ข้อมูลเพียงครึ่งหนึ่งของปริมาณข้อมูลทั้งหมด (โดย DWT อันดับที่ 1)

สาขาวิชาวิศวกรรมโทรคมนาคม	ลายมือชื่อนักศึกษา________________________
ปีการศึกษา 2553	ลายมือชื่ออาจารย์ที่ปรึกษา__________________
	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม_______________

SUPAKIT SIRIPANADORN : ANOMALY DETECTION IN WIRELESS
SENSOR NETWORKS FOR AGRICULTURE MONITORING.
THESIS ADVISOR : ASST. PROF. WIPAWEE HATTAGAM, Ph.D.,
98 PP.

WIRELESS SENSOR NETWORKS (WSNs)/AGRICULTURE MONITORING/
ANOMALY DETECTION/DISCRETE WAVELET TRANSFORM (DWT)/
SELF-ORGANIZING MAP (SOM)

Wireless Sensor Networks (WSNs) have been developed and extensively applied in agriculture monitoring. WSNs can be used to monitor and collect various physical attributes within a specific area or environment of interest. Therefore, WSNs can be viewed as a large database whose data readings from the sensors may be abnormal due to faulty sensors or unusual phenomenon in the monitored domain. However, with huge amount data, much energy is wasted in transmitting all of the measured data to the base station. Hence, in order to reduce energy consumption of transmitting all data, the data should be preprocessed prior to transmission while still maintaining the acceptable anomaly detection rate.

The underlying aim of this research is therefore to propose an anomaly detection algorithm which is able to detect anomalies accurately by means of reducing wasted energy caused by transmitting all measurement data for anomaly detection at the base station. The contribution of this research centers on the anomaly detection using Self-Organizing Map and Discrete Wavelet Transform in order to reduce the size of transmitted data without losing the significant features of the data obtained from both random number generator and collected from wireless sensor networks in a real environment.

In the experiments, the data were tested in 3 scenarios. Firstly, synthetic faults were added into synthetic and real data. The results showed that our SOMDWT algorithm can achieve true alarm rate up to 65% and 69% in case of synthetic data, and 67% and 80% in real data for the bursty and sparse faults, respectively. Note that most results had low false alarm rates, i.e., less than 1 % except in the case of sparse faults due to the increased detection difficulty. Secondly, the real faults obtained from four separate real-world datasets, namely, NAMOS, INTEL, and 2 datasets from SensorScope (pdg2008 file and SensorScope weather station no.39) were tested. The results showed that our algorithm can attain up to 99%, 100%, 83%, and 100% of true alarm rates in the NAMOS, INTEL, and SensorScope (pdg2008 file and SensorScope weather station no.39) dataset, respectively. All of the results suggested that their false alarm rates were negligible. Finally, we developed a prototype of a WSN and deployed it in a biororganic fertilizer (BOF) plant, located at the SUT university farm. The proposed SOMDWT algorithm was then tested with the real faults from the dataset acquired from the prototype. The results showed that our algorithm also performed as well as the SOM algorithm and outperformed the DWT algorithm by up to 75%. All of the results demonstrated that our proposed algorithm can maintain acceptable anomaly detection accuracy while using just half of the input data (using level 1 DWT).

School of Telecommunication Engineering

Academic Year 2010

Student’s Signature________________

Advisor’s Signature________________

Co-advisor’s Signature_____________

ACKNOWLEDGMENT

I would like to express my sincere thanks to my thesis advisors, Asst. Prof. Dr. Wipawee Hattagam and Assoc. Prof. Dr. Neung Teaumroong for their invaluable help and constant encouragement throughout the course of this research. I am most grateful for their teaching and advice, not only the research methodologies but also many other methodologies in life. I would not have achieved this far and this thesis would not have been completed without all the support that I have always received from them.

In addition, I am grateful for the lecturers in School of Telecommunication Engineering for their suggestion and all their help.

I would also like to thank Assoc. Prof. Dr. Kitti Attakitmongcol, Asst. Prof. Dr. Peerapong Uthansakul, Asst. Prof. Dr. Rangsan Tongta and Dr. Somsak Vanit-Anunchai for accepting to serve in my committee.

I am grateful for Assoc. Prof. Dr. Boon Hee Soong and Mr. Edwin Chan of Intelligent Systems Laboratory, Nanyang Technological University, Singapore for their assistance and suggestions on the sensor development and implementation.

My special thanks go to Ms. Yuppayong Junkham for her full support.

My sincere appreciation goes to Ms. Maneerat Tumpong and Ms. Pranitta Arthan for their valuable administrative support during the course of my dissertation. Finally I am most grateful to my parents and my friends both in both masters and doctoral degree courses for all their support throughout the period of this research.

Supakit Siripanadorn

TABLE OF CONTENTS

Page

ABSTACT (THAI) I

ABSTACT (ENGLISH) III

ACKNOWLEDGEMENTS V

TABLE OF CONTENTS VI

LIST OF TABLES IX

LIST OF FIGURES X

SYMBOLS AND ABBREVIATIONS XIII

CHAPTER

I INTRODUCTION 1

- 1.1 Significance of problem 1
- 1.2 Research objective 8
- 1.3 Assumption 8
- 1.4 Scope of the research 8
- 1.5 Expected usefulness 9
- 1.6 Synopsis of thesis 9

II BACKGROUND THEORY 10

- 2.1 Introduction 10
- 2.2 Anomaly detection methodology 14

TABLE OF CONTENTS (Continued)

Page

2.2.1 Univariate anomaly detection test 15
2.2.2 Multivariate anomaly detection test 15
2.3 Self-Organizing Map 16
2.4 Discrete Wavelet Transform 19
2.4.1 Haar wavelet 21
2.4.2 Daubechies wavelet 23
2.5 Integration of SOM and DWT 26
2.6 Global and Local anomaly detection 26
2.7 Summary 29
III EXPERIMENTS AND RESULTS 31
3.1 Evaluation on detecting synthetic faults 31
3.2 Evaluation on detecting faults in real-world datasets 42
3.2.1 NAMOS dataset 43
3.2.2 INTEL dataset 47
3.2.3 SensorScope dataset 50
3.3 Evaluation on detecting faults in bioorganic fertilizer plant 56
3.4 Summary 65
IV CONCLUSION AND FUTURE WORK 66
4.1 Conclusion 66

TABLE OF CONTENTS (Continued)

Page

4.1.1 Synthetic faults experiment 66
4.1.2 Faults in real-world dataset 67
4.1.3 Faults in bioorganic fertilizer plant 68
4.2 Future work 69
4.2.1 Increasing DWT level 69
4.2.2 Exploring other types of wavelets 69
4.2.3 Implementation on the sensor nodes 69
4.2.4 Comparison with other data compression techniques 69
4.2.5 Enhancing to fault predictability 70
REFERENCES 71
APPENDIX A. List of Publications 79
BIOGRAPHY 98

LIST OF FIGURES

Figure **Page**

2.1 An illustration of the SOM 18

2.2 Haar function 22

2.3 Daubechies4 function 24

2.4 The integration of the SOM and DWT algorithm diagram 29

3.2 True alarm rates with different size of neurons in the sparse faults case 35

3.3 True alarm rates with different size of neurons in the bursty faults case 35

3.4 True alarm rates with synthetic data 37

3.5 True alarm rates with real data 38

3.6 False alarm rates with synthetic data 39

3.7 False alarm rates with real data 39

3.8 True alarm rates with 10 dBW AWGN faults 41

3.9 False alarm rates with 10 dBW AWGN faults 41

3.10 Histogram shape with N = 100 (a) and N = 1000 (b) 43

3.11 NAMOS dataset of 10^4 samples 45

3.12 Detection rate in the NAMOS dataset using training epoch of 50 iterations 46

LIST OF FIGURES (Continued)

Figure **Page**

3.13 Detection rate in the NAMOS dataset using training epoch of 10 iterations 46

3.14 Detection rate in the NAMOS dataset using the size of neurons of 10x10 47

3.15 INTEL dataset of $2x10^4$ samples 48

3.16 Detection rate in the INTEL dataset using training epoch of 50 iterations 49

3.17 Detection rate in the INTEL dataset using training epoch of 10 iterations 49

3.18 Detection rate in the INTEL dataset using the size of neurons of 10x10 50

3.19 SensorScope39 dataset of 32000 samples 51

3.20 Detection rate in the SensorScope weather station no.39 dataset using training epoch of 50 iterations 52

3.21 Detection rate in the SensorScope weather station no.39 dataset using training epoch of 10 iterations 52

3.22 Detection rate in the SensorScope weather station no.39 dataset using the size of neurons of 10x10 53

3.23 SensorScope pdg dataset of 4000 samples 54

LIST OF FIGURES (Continued)

Figure **Page**

3.24 Detection rate in the SensorScope pdg2008 dataset using training epoch of 50 iterations 55
3.25 Detection rate in the SensorScope pdg2008 dataset using training epoch of 10 iterations 55
3.26 Detection rate in the SensorScope pdg2008 dataset using the size of neurons of 10x10 56
3.27 Prototype post 57
3.28 MIB520CB (USB interface board) 58
3.29 Sensor node model MPR2400 or MicaZ 59
3.30 Data acquisition board model MDA300 59
3.31 Temperature sensor (Thermocouple) 60
3.32 Moisture sensor (EC-5) 60
3.33 Bioorganic fertilizer dataset of 10740 samples 62
3.34 Detection rate in the BOF dataset using training epoch of 50 iterations 62
3.35 Detection rate in the BOF dataset using training epoch of 10 iterations 63
3.36 Detection rate in the BOF dataset using the size of neurons of 10x10 63

LIST OF TABLE

Table **Page**

3.1 Accuracy results obtained by feeding synthetic input data to the 30×30 neuron SOM algorithm. 34

SYMBOLS AND ABBREVIATIONS

SUT	Suranaree University of Technology
WSN	Wireless sensor network
BOF	Bioorganic fertilizer
KPIs	Key performance indicators
IP	Internet protocol
WANs	Wide area network
BS	Base station
SOM	Self-organizing map
LLSE	Linear least-squares estimation
ARIMA	Autoregressive integrated moving average
HMM	Hidden Markov model
DWT	Discrete wavelet transform
FDI	Fault detection and isolation
IDS	Intrusion detection system
μ	Observation index
n	The number of parameter types or key performance indices (KPIs)
x^{new}	A new state vector
K	Decision threshold
C_x	The sample covariance matrix of the data
i	Neuron
$\mathbf{m}_i$	Weight vector of neuron i

SYMBOLS AND ABBREVIATIONS (Continued)

BMU	Best matching unit
$\mathbf{x}$	Sample vector from the input data
$\lVert\cdot\rVert$	The Euclidian distance
t	The iteration index
η_t	The learning rate
$h_c(i,t)$	The neighborhood function
$r_i(t)$	The positions of neurons i
$r_c(t)$	The positions of the BMU, c
f(t)	The signal or function
ℓ	An integer index for the finite sum
a_ℓ	The real-valued expansion coefficients
$\psi_\ell(t)$	A set of real-valued functions of *t* called the expansion set
$a_{j,k}$	The discrete wavelet transform (DWT) of *f(t)*
a_j^{DWT}	The rough-scale (or approximation) coefficients
h_0	Wavelet function
g_0	Scaling function
a_{j+1}^{DWT}	A coarser set of approximation coefficients
d_{j+1}^{DWT}	A set of fine-scale (or *detail)* coefficients
$\mathrm{W}_{\mathrm{N/2}}^{1}$	The 1-level of wavelet
$\mathrm{V}_{\mathrm{N/2}}^{1}$	The 1-level of scaling signals

SYMBOLS AND ABBREVIATIONS (Continued)

$\mathbf{d}^1$	The detail coefficients for the first fluctuation subsignal
$\mathbf{a}^1$	The approximate coefficients for the first trend
N	The signal length
β	Wavelet number
α	Scaling number
E^{μ}	The error vector
N(0,1)	Normal distribution
AWGN	White Gaussian noise
n/s	The amount of inserted faults
n	The amount of faults per series
s	The amount of series of faults
T_w	The decision threshold of wavelet
$\overline{d}_1$	The sample mean of level 1 detail coefficients
db4	Daubechies4 wavelet
LP	Low pass coefficients
HP	High pass coefficients

CHAPTER I

INTRODUCTION

1.1 Significance of the Problem

Agriculture faces many challenges, for example, environmental problems such as climate changes, and water shortage; human problems such as labor shortage and usage of chemical substances, and social problems such as animal welfare and food safety. To prevent and alleviate such problems and even increase yield, condition monitoring is crucial for agriculture, particularly for crop production and farming industry. Agriculture monitoring can be applied in livestock and dairy productions to constantly monitor human food supply chain (Kwong et al., 2009). Such application allows us to monitor animal health and explore their behaviors, which is much needed to understand how the environmental condition affects animal health (Guo et al., 2006). For crops, agriculture monitoring can be applied in several stages of crop growth. In each stage, different parameters can be monitored. Effective monitoring at different stages can help save cost by reducing usage of resources such as water, pesticides, and fertilizer (Goh, Sim, and Ewe., 2007). Precision agriculture such as a hydroponics greenhouse requires precise proportion of nutrient solutions and environment conditions to achieve a suitable ecosystem which accelerate crop growth and yield. The hydroponics plants are highly sensitive to nutrient changes in the system. To achieve an efficient control in a greenhouse environment, an adaptive, accurate, cost effective control and monitoring system is needed (Li, Deng, and Ding, 2008).

Agriculture monitoring can also be applied in bioorganic fertilizer (BOF) production plants. A prototype of such plants has been constructed in Suranaree University of Technology (SUT), Thailand, to reduce production time and enhance quality control in the composting process. However, the prototype plant itself still relies mainly on manually measuring and controlling the composting parameters such as moisture, homogeneity, temperature, pH, oxygen, soil nutrients, etc., which is both time consuming and laborious. Autonomous monitoring devices such as wireless sensor networks therefore warrant potential use in the composting process.

A wireless sensor network (WSN) is a wireless network that consists of distributed autonomous sensoring devices which cooperatively monitor or collect environmental conditions such as temperature, sound, vibration, pressure, motion, or pollutants at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, WSNs are now used in many civilian application areas such as environment and habitat monitoring, healthcare monitoring applications, and traffic control.

There are several measurements which can be collected from a WSN deployed in BOF plants. These measurements are vital to the BOF production. In other words, these measurements can be viewed as key performance indicators (KPIs) in the fertilizer plant. A good choice of a set of KPIs to monitor and analyze the collected data is crucial to understanding the reasons for the various operational states of the WSN, noticing abnormal incidents, analyzing them and providing a suitable course of action. With the huge amount of data continually collected from the WSN, it becomes increasingly difficult to detect anomalies in the data measurements.

Therefore, anomaly detection techniques are necessary to automatically detect actual faults and alert the system controller to take a suitable action.

Research emphasizing on anomaly detection in communication networks has progressed in recent years. These works include a statistical analysis approach in IP networks that focused on load anomaly detection in segments of IP networks that carry (almost) exclusively voice traffic (Thottan and Chuanyi, 2003); (Hajji, 2003); (Ho et al., 2000). proposed a service anomaly detection algorithm in wide area networks (WANs) which generated alarms upon detecting exceptional states such as a router interface being “down” or the utilization of a network segment exceeding a predefined threshold. Ref. (Laiho et al., 2005, 2002); (Barreto et al., 2000). investigated anomaly detection in cellular mobile networks which required monitoring hundreds of adjustable variables in each cell consisting of parameters within the base stations (BS) and quality information of the calls. Ref. (Feather, Siewiorek, and Maxion, 1993); (Maxion and Feather, 1990). studied anomaly detection for two types of failures in Ethernet segments, namely, hard failures which are characterized by the inability to deliver packets, and soft failures which are characterized by a partial loss of network bandwidth. Possible causes of a hard failure include power failure, cut cable, or failure of major network equipment. Causes of soft failures include inappropriate use of the network, temporary congestion causing delayed transmission, failed host hardware, failure of higher level protocols, or mischievous users. Ref. (Hood and Ji, 1997); (Hood and Ji, 1997). proposed anomaly and performance change detection algorithms in small networks that improved network reliability and management in high-speed communication networks. In particular, they proposed an intelligent system using adaptive statistical approaches to learn the normal behavior of the network. Deviations from the norm were detected and the information was combined in the probabilistic framework of Bayes network. Their method can thereby detect unknown or unseen faults. As demonstrated on real network data, their method can detect abnormal behavior before a fault actually occurs, giving the network management system the ability to avoid a potentially serious problem. Ref. Sukkhawatchani, P., and Usaha, W. (2008). also investigated anomaly detection with real network data. They applied a competitive learning algorithm called self-organizing map (SOM) to detect traffic measurement anomalies from an actual cellular network service provider.

There are also works on fault and anomaly detection in wireless sensor networks (WSNs). In general, anomaly detection in WSNs refers to the problem of finding patterns in data that do not conform to expected behavior (Kaur, Saxena, and Gupta, 2010). Abnormal data patterns can be caused by faulty sensors in the network or unusual phenomena in the monitored domain.

Anomalies caused by faulty sensor communications were presented in (Lee and Choi, 2008). They proposed a distributed algorithm for detecting and isolating faulty sensor nodes in WSNs. Each sensor node identified its own status based on local comparisons of sensed data with thresholds. Ref. (Sharma, Golubchik, and Govindan, 2010). applied 4 different anomaly detection techniques, e.g., the rule-based method, the linear least-squares estimation (LLSE) method, the autoregressive integrated moving average (ARIMA) method, and the learning-based hidden Markov model (HMM) method, for different types of faults obtained in the real–world datasets, namely, NAMOS (NAMOS, 2006). INTEL (INTEL, 2006). and SensorScope (SensorScope, 2006). They classified these faults into 3 types, i.e., noise faults, short faults, and constant faults. Their research suggested that there is presently no known anomaly detection method suitable for every type of fault.

Another application of anomaly detection is to detect an unusual phenomenon in the monitored domain. Erroneous measurements may occur as a result of transducers, or from faults introduced by harsh environmental conditions. In a large network, it is extremely difficult and time consuming to detect these erroneous measurements manually. In addition, energy is wasted in the network when forwarding the unwanted erroneous measurements to the base station for analysis. One solution to alleviate network energy consumption is to reduce the amount of data that needs to be communicated through the network. As energy expenditure is critical in WSNs, anomaly detection methods in WSN must not only perform well but also demand low energy consumption. Distributed in-network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN (Rajasegarar, Leckie, and Palaniswami, 2008); (Cordina and Debono, 2008). focused on increasing the lifetime of a WSN using an approach that relies on cluster-based routing algorithms. The lifetime of the sensor network was improved through the use of a number of mechanisms that minimized energy dissipation and improved energy balancing between the nodes. These mechanisms included cluster head separation, cluster head election, cluster head rotation, and load balancing cost functions. Our work was motivated by the concept of prolonging the network lifetime. In particular, we focused on finding means to reduce the amount of transmitted data in the network for anomaly detection at the base station, while still maintaining acceptable accuracy and reliability in detecting abnormal data.

This thesis considered anomalies caused by unusual phenomenon and faulty sensors. To detect these anomalies, a dynamic data classification scheme such as data mining method could be useful. ta mining is an expanding area of research in artificial neural network and information management whose objective is to extract relevant information from large databases. Typical data mining and analysis tasks include classification, regression, and clustering of data, aiming at determining parameter/data dependencies and finding various anomalies from the data. One particular method, called the self-organizing map (SOM), has several beneficial features which make it a useful tool in data mining. In particular, it follows the probability density function of the data and is, thus, an efficient clustering and quantization algorithm. The most important feature of the SOM in data mining is the visualization property (Laiho et al., 2005)

SOM has been applied for anomaly detection in communication networks (Barreto et al., 2006); (Sukkhawatchani, P., and Usaha, W. (2008)); (Zheng and Hu, 2005). as well as WSNs (Paladina, Paone, Jellamo, and Puliafito, 2007). focused on evaluating the position of sensors in a WSN, or the localization problem. Their localization technique was based on a simple SOM, implemented on each sensor node. The main advantages of their solution were the limited storage and computing costs. However, SOM requires processing time which increases with the size of input data. To reduce the input data size, features of the data can be extracted without losing the significant data which can be used for anomaly detection. This can be achieved by the Discrete Wavelet Transform (DWT).

Wavelets have been extensively employed for anomaly (Aquino and Barria, 2001). and fault detection (Yadaiah and Ravi, 2007). DWT had also been integrated with SOM for several applications. Ref. (Doshi, King, and Lawrence, 2007). used DWT and SOM for feature extraction for nematode species identification. Hyperspectral data, which typically has high dimensions, was used for species identification. DWT and SOM provided the necessary dimensionality reduction without losing vital information. In (Xu and Zhao, 2002). DWT and SOM were employed for fault detection and isolation (FDI) to control a tank system. In order to detect the faults that reflected themselves as fault-induced frequency changes at certain time instants in the measured signal, DWT was applied to capture such changes and extract fault features online and in real-time. An improved SOM was then used to isolate the fault. In Ref (Postalcıoglu, Erkan, and Bolat, 2007). multiplicative and additive types of sensor faults had been examined and disturbance had been applied to create faults in temperature sensors. In particular, feature vectors of the sensor faults had been constructed using DWT, sliding window, and a statistical analysis. Classification of the feature vectors was obtained by using SOM.

To the best of our knowledge, the integration of DWT and SOM has not yet been applied for anomaly detection in WSNs. Therefore, the underlying aim of this paper was to propose an anomaly detection algorithm which determined the discrete wavelet transform, and detects the abnormality of the sensor readings by training the SOM using the wavelet coefficients. Our proposed algorithm, the integrated SOM and DWT algorithm could help reduce wasted energy caused by transmitting all measurement data to the base station by applying DWT algorithm onto the sensor modes in order to reduce size of transmitted data without losing the significant feature of the data.

1.2 Research Objectives

The objectives of this thesis are as follows:

1.2.1 To obtain the anomaly detection algorithm for a wireless sensor network which can reduce the energy consumption required for transmitting data packets back to the base station.

1.2.2 To obtain a wireless sensor network prototype for a BOF production plant.

1.3 Assumptions

1.3.1 Abnormal data which occur in a wireless sensor network can be detected by changing signal levels collected from the sensors.

1.3.2 Faults can be caused by faulty sensors in the network or unusual phenomena in the monitored domain.

1.4 Scope of the Research

1.4.1. A wireless sensor network prototype was designed for agriculture monitoring.

1.4.2. Anomaly detection methods for WSNs were studied.

1.4.3. Discrete wavelet transform (DWT) and self-organizing map (SOM) were studied to detect abnormal data collected from a wireless sensor network in an agriculture field.

1.4.4. The simulation and experimental findings were analyzed and concluded.

1.5 Expected Usefulness

1.5.1 To obtain an algorithm implemented at the base station or gateway which detects abnormal data collected from a wireless sensor network.

1.5.2 To obtain a wireless sensor network prototype for a BOF production plant.

1.6 Synopsis of Thesis

The remainder of this thesis is organized as follows. **Chapter 2** introduces the theoretical background which is the foundation of the contributions of this thesis. Firstly, the concept of the general anomaly detection and related works are presented. This is followed by the self-organizing map (SOM) algorithm, the discrete wavelet transform (DWT) algorithm and the proposed algorithm, the integrated SOM and DWT.

Chapter 3 presents the experiments conducted to evaluate the performance of the proposed integration of SOM and DWT algorithm. The experiments evaluated the anomaly detection methods described in the previous chapter with series of synthetic data and actual data collected from a wireless sensor network injected by various synthetic faults. Furthermore, its performance against real-world datasets which included real faults from various sensor networks were also evaluated.

Finally, **Chapter 4** summarizes all the findings and the original contribution in this thesis and points out possible future research directions.

CHAPTER II

BACKGROUND THEORY

2.1 Related works

In anomaly detection scenarios, several methods have been presented. For example, Rule-based methods such as the histogram method (Ramanathan et al., 2006) require domain knowledge about sensor readings to develop heuristic rules or constraints that the sensor readings must satisfy. Although methods belonging to this group can be highly accurate, the choice of parameters is critical to their accuracy; Estimation-based methods such as the linear least square estimation (LLSE) Kailath (1977) define "normal" sensor behavior by using spatial correlation from measurements at different sensors. This method is accurate, but cannot classify faults; Time series analysis-based methods such as the autoregressive integrated moving average (ARIMA) (Box, Jenkins, and Reinsen, 1994) are commonly used for analyzing periodically and temporally correlated data collected by the same sensor. However, such methods are more effective for detecting short duration faults than long duration ones, and incur more false alarm than the other methods; Learning-based methods such as the Hidden Markov model (HMM) Bengio and Frasconi (1996) or neural networks model the normal and faulty sensor readings using training data and are therefore suitable for phenomena that may not be spatio-temporally correlated. Learning methods can be cumbersome to train, but can accurately detect anomalies (Sharma, Golubchik, and Govindan, 2010). The proposed algorithm in this thesis is based on a learning method using self- organizing map (SOM). SOM has several beneficial features which make it a useful tool in data mining. In particular, it follows the probability density function of the data and is, thus, an efficient clustering and quantization algorithm. The most important feature of the SOM in data mining is the visualization property (Laiho et al., 2005)

This chapter serves as an introduction to related anomaly detection methods which were used to improve our algorithm, the integrated SOM and DWT, and compared with our algorithm. These methods include the single threshold (univariate and multivariate) methods (Barreto et al., 2006) the self-organizing map (SOM) algorithm and the discrete wavelet transform (DWT).

Data acquisition in wireless sensor networks (WSNs) involves intensive collection of input data from sensor nodes. Algorithms based on data mining had proved to be especially suitable in highly complex and data intensive applications (Laiho et al., 2005).The self-organizing map (SOM) is one of the most popular neural algorithms due to its efficient visualization properties. It has been used for anomaly detection in various scenarios. Ref. (Sukkhawatchani, P., and Usaha, W., 2008) used the SOM algorithm to analyze and monitor traffic anomalies in a cellular mobile network. Their results showed that SOM outperformed the multivariate method. Furthermore, the SOM can be able to handle a larger number of the interested or collected data.

Apart from cellular networks, (Ramadas, Ostermann, and Tjaden, 2003) proposed an anomaly detection system based on SOM algorithm at level of network connection, where each connection was defined with six features: duration of the connection, type of protocol, type of service, state of the connection, source bytes and destination bytes. Ref. (Lichodzijewski, Heywood, and Heywood, 2002) described the use of SOM in the intrusion detection system (IDS) to detect anomalies in the network connections by characterizing each connection with six statistical features. However, both works used a different set of the features. Another difference between these two works was that in the first work SOM was applied to all the network traffic, whereas the latter constructed a neural network by each network service.

Anomaly detection in WSNs, however, requires careful consideration than other applications because energy expenditure is critical in such networks. Therefore, anomaly detection methods in WSN must not only perform well but also demand low energy consumption. One solution to alleviate network energy consumption is to reduce the amount of data that needs to be communicated through the network. Wavelet analysis technique was considered for dimensionality reduction. Processing of the wavelet coefficients may allow signal representation with a lower number of bits than needed for representing the original signal (Brechet, Lucas, Doncarli, and Farina, 2007).

The wavelet analysis technique has also been widely used for network anomaly detection recently due to its inherent time-frequency property that allows splitting signals into different components at several frequencies. In the work (Barford, Kline, Plonka, and Ron, 2002) wavelet transform was applied for analyzing and characterizing the flow-based traffic behaviors. Based on different frequency components, a deviation algorithm was presented to identify anomalies by setting a threshold for the signal composed from the wavelet coefficients at different frequency levels. Ref. (Kim, Reddy, and Vannucci, 2004) proposed a technique for traffic anomaly detection through analyzing correlation of destination IP addresses in outgoing traffic at a router. They applied discrete wavelet transform on the address and port number correlation data over several time scales. Any deviation from historical regular norms would alert the network administrator of the potential anomalies in the traffic.

Wavelet was not only applied for detecting specific network anomalies directly, it was also widely used in network measurement from the perspectives of traffic performance analysis (Huang, Feldmann, and Willinger, 2001) traffic anomalies diagnosing and mining (Lakhina, Crovella, and Diot, 2004) traffic congestion detection (Kim et al., 2004) and dimensionality reduction (Bruce, Cheriyadat, and Burns, 2003); (Bruce, Koger, and Li, 2002); (Ciancio, Pattem, Otega, and Krishnamachari, 2006) addressed the problem of energy consumption reduction in wireless sensor networks by means of data compression using lifting factorization wavelet transform. In particular, they reconstructed a version of the data measurements at the central node, with the sensors spending as little energy as possible, for a given data reconstruction accuracy. Ref (Li, Zhang, and Fang, 2009) also proposed a data compression algorithm in WSN based on lifting wavelet transform. Their algorithm distributed the computing quantity which the lifting wavelet transform required to all nodes, eliminated extra computing and data transmission, thereby reduced the information redundancy of network, and consequently saved the energy of wireless transmission and prolonged network lifetime.

The anomaly detection of SOM and the data reduction ability in wavelets therefore motivated us to integrate these two methods to achieve efficient and effective anomaly detection methods for condition monitoring. To the best of our knowledge, there is no prior work applying such combination for anomaly detection in WSNs. In a similar work such as (Xu and Zhao, 2002). DWT and SOM were employed for fault detection and isolation (FDI) in the control system (tank system). There were existing works combining SOM and DWT for classification application as in a fault tolerant control system (Postalcıoglu, Erkan, and Bolat, 2007) where the feature vectors of the sensor faults have been constructed using DWT, sliding window and a statistical analysis. Classification of the feature vectors was obtained by using SOM, clustering application. In (Cheng, Zhang, Hu, and Li, 2007) they proposed the combination of the DWT and SOM algorithm to cluster the urban traffic flow network. Discrete wavelet transform was adopted for flow feature extraction, because it is insensitive to disturbance/scaling, and zooms in multiple finer granularities. The self-organizing map (SOM) algorithm was then used to cluster road links into groups, for which different feature sets were considered for different purposes.

2.2 Anomaly Detection

Network anomaly refers to circumstances when network operations deviate from normal network behavior. Abnormal data patterns can be caused by faulty sensors in the network or unusual phenomena in the monitored domain. This thesis, we are interested in monitoring anomaly detection in the data gathered from WSNs.

The first step of anomaly detection involves selecting the data parameters to be monitored and grouping them together in a pattern vector $\mathbf{x}^{\mu} \in \Re$, $\mu = 1, \ldots, N$,

$$x^{\mu} = \begin{pmatrix} x_1^{\mu} \\ x_2^{\mu} \\ \vdots \\ x_n^{\mu} \end{pmatrix} = \begin{pmatrix} \mathrm{KPI}_1^{\mu} \\ \mathrm{KPI}_2^{\mu} \\ \vdots \\ \mathrm{KPI}_{\mathrm{n}}^{\mu} \end{pmatrix} \tag{2.1}$$

where μ is the observation index, n is the number of parameter types or key performance indices (KPIs) chosen to monitor the environmental condition.

The second step involves identifying the methodology used to classify a new state vector x^{new} as normal or abnormal. Single threshold methods, such as the univariate and the multivariate anomaly detection tests are currently deployed by means of one of the following statistical methods.

2.2.1 Univariate Anomaly Detection Test

A component-wise analysis is performed on x^{new} to verify if the components x_j^{new} are within their normal ranges of variation. The anomaly detection test is given by

$$\text{IF } \left| \frac{\mathrm{x}_j^{new} - \overline{\mathrm{x}_j}}{\sigma_j} \right| < \mathrm{K}$$

$$\text{THEN } \mathrm{x}_j^{new} \text{ is a NORMAL component}$$

$$\text{ELSE } \mathrm{x}_j^{new} \text{ is an ABNORMAL component} \tag{2.2}$$

where the decision threshold K > 0, and the normal range interval of x_j^{new} is $[\overline{\mathrm{x}_j} - \mathrm{K}\sigma_j, \overline{\mathrm{x}_j} + \mathrm{K}\sigma_j]$. Typically, the value of K relies on the Gaussian distributed $\mathbf{x}_j$, e.g., K = 1.96 for confidence interval 95% or K = 2.57 for confidence interval 99%.

2.2.2 Multivariate Anomaly Detection Test

The analysis is performed on x^{new} as a whole, by taking into consideration the joint influence of all the components x_j^{new}, $j = 1,\ldots,n$. The anomaly detection test is given by

$$\text{IF } \left\{ \left(\mathrm{x}^{new} - \overline{\mathrm{x}} \right)^T C_x^{-1} \left(\mathrm{x}^{new} - \overline{\mathrm{x}} \right) \right\}^{\frac{1}{2}} < \mathrm{K}$$

THEN x^{new} is a NORMAL vector

ELSE x^{new} is an ABNORMAL vector (2.3)

where $C_x = \left(\frac{1}{N} \right) \sum_{\mu=1}^{N} \left(\mathrm{x}^{\mu} - \overline{\mathrm{x}} \right) \left(\mathrm{x}^{\mu} - \overline{\mathrm{x}} \right)^T$ is the sample covariance matrix of the data, and K is the critical value $X_{n,1-\alpha}^2$ of the chi-square distribution with *n* degrees of freedom and significance level α.

These methods are based on the assumption that the components of **x** are Gaussian distributed. Their drawbacks are that different detection results are obtained when outliers are present in the data set. This is because outlier distorts the estimates of $\overline{x}$ and C. Therefore, anomalies with slight departures from normal state or base line are difficult to detect.

2.3 Self-Organizing Map

Competitive neural models such as the self-organizing map (SOM) (Laiho, Kylvaja, and Hoglund, 2002) are able to extract statistical regularities from the input data vectors and encode them in the weights without supervision. It maps a high-dimensional data manifold onto a low-dimensional, usually two-dimensional, grid or display.

The basic SOM consists of a regular grid of map units or neurons as shown in Figure 2.1(a). Each neuron, denoted by i (depicted by the black dot), has a set of layered neighboring neurons (depicted by the white dots) as shown in Figure 2.1(a).

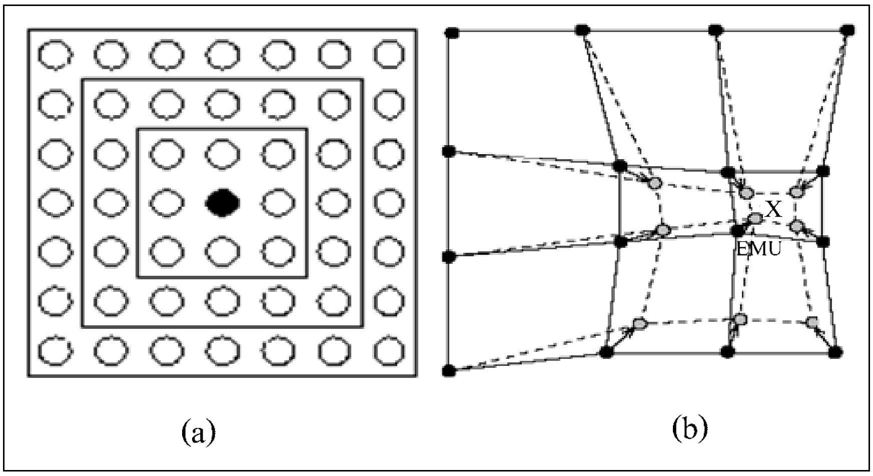
Neuron *i* maintains a weight vector $\mathbf{m}_i$. In order to follow the properties of the input data, such vector is updated during the training process. For example, Figure 2.1 (b) shows a SOM represented by a 2-dimensional grid of 4×4 neurons. The dimension of each vector is equal to the dimension of the input data. In the figure, a vector of input data (marked by x) is used to train the SOM weight vectors (the black dots). The winning neuron (marked by BMU) as well as its 1-neighborhood neurons, adjusts their corresponding vectors to the new values (marked by the gray dots).

The SOM is trained iteratively. In each training step, one sample vector $\mathbf{x}$ from the input data set is chosen.

The distances between the sample data and all of weight vectors in the SOM are calculated using some distance measure. Suppose that at iteration *t*, neuron *i* whose weight vector $\mathbf{m}_i(t)$ is the closest to the input vector $\mathbf{x}(t)$. We denote such weight vector by $\mathbf{m}_c(t)$ and refer to it as the Best-Matching Unit (BMU), which is

$$\left\| x(t) - m_c(t) \right\| = \arg \min_{\forall i} \left\| \mathbf{x}(t) - \mathbf{m}_i(t) \right\| \tag{2.4}$$

where $\|\cdot\|$ is the Euclidian distance.



(a) (b)

Figure 2.1 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3 neighborhoods, (b) SOM updates the BMU with 1- neighborhood.

Suppose neuron i is to be updated, the SOM updating rule for the weight vector of neuron i is given by

$$m_i(t+1) = m_i(t) + \eta_t h_c(i,t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{2.5}$$

where t is the iteration index, $\mathbf{x}(t)$ is an input vector, η_t is the learning rate, $h_c(i,t)$ is the neighborhood function of the algorithm. The Gaussian neighborhood function may be used, that is

$$h_c(i,t) = \exp\left(-\frac{\left\| r_c(t) - r_i(t) \right\|^2}{2\sigma^2(t)} \right) \tag{2.6}$$

where $r_i(t)$ and $r_c(t)$ are the positions of neurons i and the BMU c respectively, and $\sigma(t)$ is the radius of the neighborhood function at time t. Note that $h_c(i,t)$ defines the width of the neighborhood. It is necessary that $\lim_{t\to\infty} h_c(i,t) = 0$ and $\lim_{t\to\infty} \eta_t = 0$ for the algorithm to converge (Barreto et al., 2006).

SOM has been applied for anomaly detection in communication networks (Barreto et al., 2006); (Sukkhawatchani, P., and Usaha, W., 2008); (Zheng and Hu, 2005) as well as WSNs (Paladina, Paone, Jellamo, and Puliafito, 2007) focused on evaluating the position of sensors in a WSN, or the localization problem. Their localization technique was based on a simple SOM, implemented on each sensor node. The main advantages of their solution were the limited storage and computing costs. However, SOM requires processing time which increases with the size of input data. To reduce the input data size, features of the data can be extracted without losing the significant data can be used for anomaly detection. This can be achieved by the discrete wavelet transform (DWT).

2.4 Discrete Wavelet Transform

Wavelet is a small wave which has its energy concentrated in time to give a tool for the analysis of transient, non-stationary, or time-varying phenomena. The goal of the wavelet analysis is to create a set of basis functions and transforms that will give an informative, efficient, and useful description of a function or signal. If the signal is represented as a function of time, wavelets provide efficient localization in both time and frequency or scale. If signal or function *f(t)* often be better analyzed, described, or processed if expressed as a linear decomposition by

$$f(t)=\sum_{\ell} a_{\ell}\psi_{\ell}(t) \tag{2.7}$$

where ℓ is an integer index for the finite sum, a_{ℓ} are the real-valued expansion coefficients, and $\psi_{\ell}(t)$ are a set of real-valued functions of *t* called the expansion set. For the wavelet expansion, a two-parameter system is constructed such that (2.7) becomes

$$f(t)=\sum_{k}\sum_{j} a_{j,k}\psi_{j,k}(t) \tag{2.8}$$

where both *j* and *k* are integer indices and the $\psi_{j,k}(t)$ are the wavelet expansion functions or mother wavelet, i.e., Haar mother wavelet, Daubechies mother wavelet or Mexican Hat mother wavelet. The set of expansion coefficients $a_{j,k}$ are called the discrete wavelet transform (DWT) of *f(t)* and (2.8) is the inverse transform.

DWT is a mathematical transform that separates the data signal into fine-scale information known as detail coefficients, and rough-scale information known as approximate coefficients. Its major advantage is the multi-resolution representation and time-frequency localization property for signals. Usually, the sketch of the original time series can be recovered using only the low-pass-cut off decomposition coefficients; the details can be modeled from the middle-level decomposition coefficients; the rest is usually regarded as noises or irregularities. The following equations describe the computation of the DWT decomposition process:

$$a_{j+1}^{DWT}(k)=\sum_{n} h_0(n-2k)a_{j}^{DWT}(k) \tag{2.9}$$

$$d_{j+1}^{DWT}(k) = \sum_{n} g_0(n-2k) a_j^{DWT}(k), \tag{2.10}$$

where the rough-scale (or approximation) coefficients a_j^{DWT} are convolved separately with h_0 and g_0, the wavelet function and scaling function, respectively, n is the time scaling index, k is the frequency translation index for wavelet level j. The resulting coefficient is down-sampled by 2. This process splits a_j^{DWT} roughly in half, partitioning it into a set of fine-scale (or *detail)* coefficients d_{j+1}^{DWT} and a coarser set of approximation coefficients a_{j+1}^{DWT} (Postalcıoglu, Erkan, and Bolat, 2007).

DWT has the capability to encode the finer resolution of the original time series with its hierarchical coefficients. Furthermore, DWT can be computed efficiently in linear time, which is important while dealing with large datasets. That is the DWT can reduce amount of the input data without losing significant feature of the data by replacing the data with its hierarchical coefficients, low pass and high pass coefficients.

In our experiment, the Haar and Daubechies4 wavelet were used as a mother wavelet in order to reduce the size of the data before performing the anomaly detection with the SOM algorithm. The reason for using these two types of wavelets is because they are relatively easy to cross-check by hand with computed coefficients from MATLAB program.

2.4.1 Haar Wavelet

Haar wavelet is the simplest type of wavelet with its unit width, unit length, unit height pulse function as shown in Figure 2.2 Like all wavelet transforms, the Haar transform decomposes a discrete signal into 2 subsignals of half its length Walker (1999).

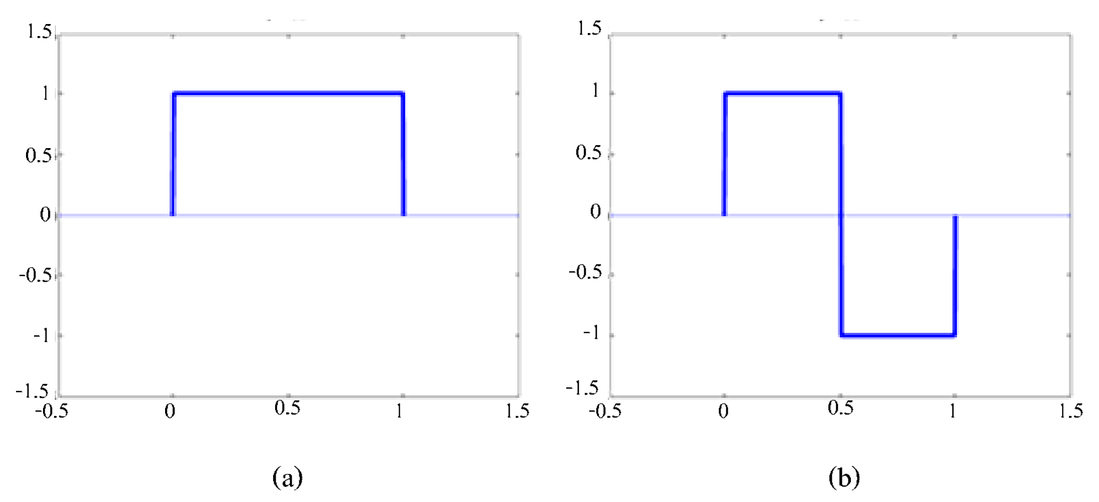


Figure 2.2 Haar scaling (a) and wavelet (b) function from http://cnx.org/content/m11150/latest

One subsignal is an averaging or *trend*; the other subsignal is a difference or *fluctuation*. Let's begin with the 1-level Haar wavelet. These wavelets are defined as

$$\begin{aligned} W_1^1 &= \left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0, \ldots, 0\right) \\ W_2^1 &= \left(0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0, \ldots, 0\right) \\ &\vdots \\ W_{N/2}^1 &= \left(0, 0, \ldots, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right) \end{aligned} \tag{2.11}$$

They all are very similar to each other in that they are each a translation in time by an even number of time-unit of the first Haar wavelet W_1^1. The second Haar wavelet W_2^1 is a translation forward in time by 2 units of W_1^1, and W_3^1 is a translation forward in time by four units of W_1^1, and so on.

They can also express the 1-level trend values as scalar products with certain elementary signals. These elementary signals are called 1-level Haar scaling signals, and they are defined as

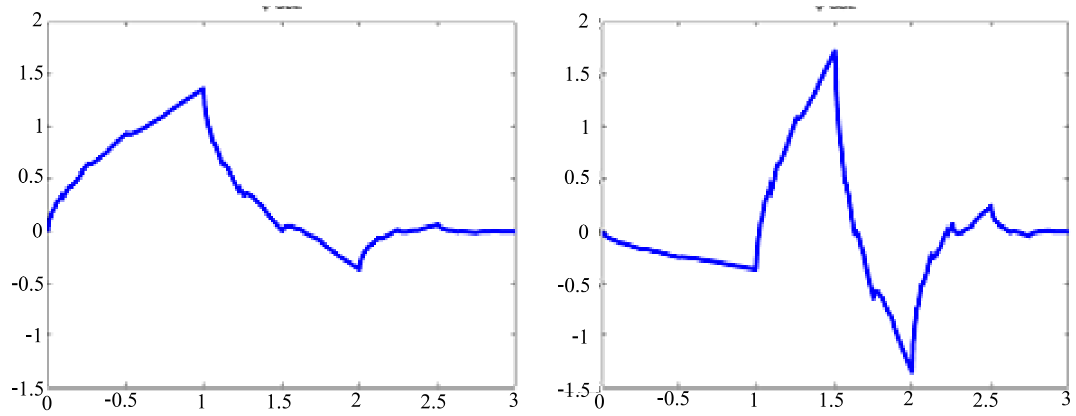
$$
\begin{aligned}
V_1^1 &= \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right) \\
V_2^1 &= \left(0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right) \qquad (2.12) \\
&\vdots \\
V_{N/2}^1 &= \left(0, 0, \ldots, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)
\end{aligned}
$$

Using these Haar wavelet, the detail coefficients for the first fluctuation subsignal $\mathbf{d}^1$ are expressed as scalar products, $d_m = f \cdot W_m^1$ and also using the Haar scaling signals, the approximate coefficients $\mathbf{a}^1$ for the first trend are expressed as scalar products $a_m = f \cdot V_m^1$ for m = 1, 2,…, N/2, where **f** is the signal and N is the signal length.

2.4.2 Daubechies Wavelet

The Daubechies wavelet transforms are defined in the same way as the Haar wavelet transform by computing averages and differences via scalar products with scaling signals and wavelets. The only difference between them consists in how these scaling signals and wavelets are defined. Let the wavelet numbers $\beta_1, \beta_2, \beta_3, \beta_4$ be defined by

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \beta_2 = \frac{\sqrt{3}-3}{4\sqrt{2}}, \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}} \tag{2.13}$$



(a) (b)

Figure 2.3 Daubechies4 scaling (a) and wavelet (b) function from http://cnx.org/content/m11150/latest

Using these wavelet numbers, the 1-level Daubechies4 wavelets are defined by

$$\begin{aligned}
W_1^1 &= (\beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \ldots, 0) \\
W_2^1 &= (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \ldots, 0) \\
W_3^1 &= (0, 0, 0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \ldots, 0) \\
&\vdots \\
W_{N/2-1}^1 &= (0, 0, \ldots, 0, \beta_1, \beta_2, \beta_3, \beta_4) \\
W_{N/2}^1 &= (\beta_3, \beta_4, 0, 0, \ldots, 0, \beta_1, \beta_2)
\end{aligned} \tag{2.14}$$

These are all translates of W_1^1. Each wavelet has a support of just 4 units, corresponding to the four non-zero wavelet numbers used to define them. The 1-level Daubechies4 wavelet satisfy

$$\mathrm{W}_{\mathrm{m}}^1=\beta_1\mathrm{V}_{2\mathrm{m}-1}^0+\beta_2\mathrm{V}_{2\mathrm{m}}^0+\beta_3\mathrm{V}_{2\mathrm{m}+1}^0+\beta_4\mathrm{V}_{2\mathrm{m}+2}^0 \tag{2.15}$$

We now turn to a discussion of the Daubechies4 scaling signals. Let the scaling numbers $\alpha_1,\alpha_2,\alpha_3,\alpha_4$ be defined by

$$\alpha_1=\frac{1+\sqrt{3}}{4\sqrt{2}},\ \alpha_2=\frac{3+\sqrt{3}}{4\sqrt{2}},\ \alpha_3=\frac{3-\sqrt{3}}{4\sqrt{2}},\ \alpha_4=\frac{1-\sqrt{3}}{4\sqrt{2}} \tag{2.16}$$

Using these scaling numbers, the 1-level Daubechies4 scaling signals are defined by

$$\begin{aligned}
\mathrm{V}_1^1&=\left(\alpha_1,\alpha_2,\alpha_3,\alpha_4,0,0,\ldots,0\right)\\
\mathrm{V}_2^1&=\left(0,0,\alpha_1,\alpha_2,\alpha_3,\alpha_4,0,0,\ldots,0\right)\\
\mathrm{V}_3^1&=\left(0,0,0,0,\alpha_1,\alpha_2,\alpha_3,\alpha_4,0,0,\ldots,0\right)\\
&\ \ \vdots\\
\mathrm{V}_{\mathrm{N/2-1}}^1&=\left(0,0,\ldots,0,\alpha_1,\alpha_2,\alpha_3,\alpha_4\right)\\
\mathrm{V}_{\mathrm{N/2}}^1&=\left(\alpha_3,\alpha_4,0,0,\ldots,0,\alpha_1,\alpha_2\right)
\end{aligned} \tag{2.17}$$

As same as the Daubechies4 wavelet, the first level Daubechies4 scaling signals satisfy

$$\mathrm{V}_{\mathrm{m}}^1=\alpha_1\mathrm{V}_{2\mathrm{m}-1}^0+\alpha_2\mathrm{V}_{2\mathrm{m}}^0+\alpha_3\mathrm{V}_{2\mathrm{m}+1}^0+\alpha_4\mathrm{V}_{2\mathrm{m}+2}^0 \tag{2.18}$$

2.5 Integration of SOM and DWT

In our proposed algorithm in this thesis, the integration of SOM and DWT, the DWT algorithm was used as an input data preprocessor of the SOM algorithm in order to reduce the size of input data without losing any significant feature of the data. This can enable the implementation of in-network processing which can help to reduce the radio communication energy and eventually prolong the lifetime of the WSN (Rajasegarar, Leckie, and Palaniswami, 2008). The input data were padded with zero if its length was odd. After obtaining the wavelet coefficients, these coefficients were fed to the SOM algorithm which can be divided into 2 sets. Each set contained both approximate and detail coefficients. The first set which was obtained from noiseless data, was used to train the SOM algorithm. The second set which was obtained from the faulty data would be used to test the SOM algorithm. Then to reduce the false alarms the detected results were double checked by using the univariate method (Barreto et al., 2006); (Sukkhawatchani, P., and Usaha, W. (2008).

2.6 Anomaly Detection

A new observation data set can be considered abnormal if the distance between the weight vector of the winning neuron and the new state vector, given by

$$e^{\mu} = \left\| \mathbf{x}^{new} - \mathbf{m}_c^{\mu} \right\| \tag{2.19}$$

is greater than a certain percentage $p = 1 - \alpha$ of the distances in the distance distribution profile. That is, IF $e^{\mu} \in \left[e_p^{-}, e_p^{+}\right]$,

THEN $\mathbf{x}^{new}$ is NORMAL vector

ELSE $\mathbf{x}^{new}$ is ABNORMAL vector (2.20)

Equation (2.20) is referred to as the global decision. Once abnormal behavior has been detected by the procedure in (2.20), it is necessary to investigate which of KPIs of the problematic state vector are the most relevant ones. That is called the local decision. We require instead a kind of decision rule that points out the KPIs (if any) that contribute most to the supposed abnormal state vector. For this purpose, we evaluate the absolute values of each component of the error vector E^{μ}

$$\left|E^{\mu}\right| = \begin{pmatrix} \left|E_1^{\mu}\right| \\ \left|E_2^{\mu}\right| \\ \vdots \\ \left|E_n^{\mu}\right| \end{pmatrix} = \begin{pmatrix} \left|x_1^{\mu} - w_{i*1}^{\mu}\right| \\ \left|x_2^{\mu} - w_{i*2}^{\mu}\right| \\ \vdots \\ \left|x_n^{\mu} - w_{i*n}^{\mu}\right| \end{pmatrix} \tag{2.21}$$

Thus, for each of the resulting *n* sample distributions $\left\{\left|E_j^{\mu}\right|\right\}, \mu = 1, \ldots, N$ we compute the interval of normality $\left[\left|E_j^{-}\right|, \left|E_j^{+}\right|\right]$ of the *j*th KPI.

Whenever an incoming state vector $\mathbf{x}^{\mathrm{new}}$ is considered abnormal by the fault detection stage, we take the absolute values of each component $\mathrm{E}_{\mathrm{j}}^{\mathrm{new}}$ of the corresponding quantization error vector and execute the following test:

$$\text{IF } \left|E_{\mathrm{j}}^{\mathrm{new}}\right| \in \left[\left|E_j^{-}\right|, \left|E_j^{+}\right|\right]$$

$$\text{THEN } x_j \text{ is a NORMAL KPI}$$

$$\text{ELSE } x_j \text{ is an ABNORMAL KPI} \tag{2.22}$$

That is, if the quantization error due to the KPI x_j is within the range defined by the interval $\left[\left|E_j^{-}\right|, \left|E_j^{+}\right|\right]$, then it does not contribute to the abnormal state previously detected; otherwise it will be indicated as an abnormal KPI. In Lee and Choi (2008) an addition of local decisions of each KPI is presented.

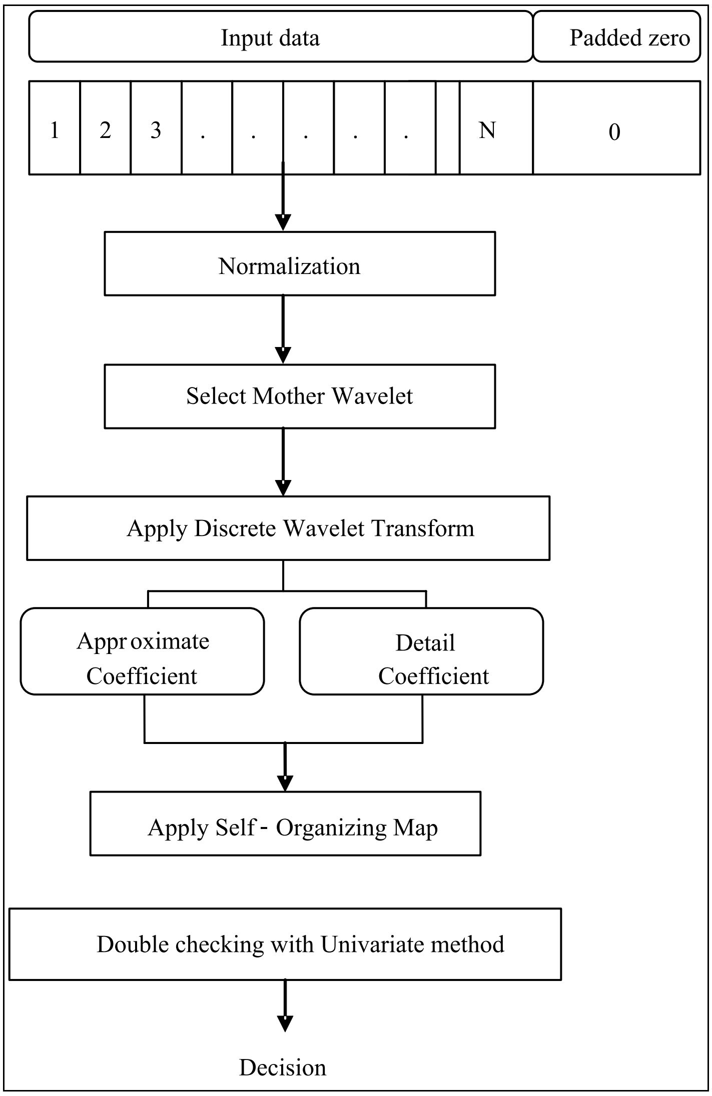


Figure 2.4 The integration of the SOM and DWT algorithm diagram.

2.7 Summary

In this chapter, we introduced the anomaly detection methods including the univariate anomaly detection, multivariate anomaly detection, the self-organizing map or SOM, the discrete wavelet transform or DWT algorithm, and the proposed integrated SOMDWT algorithm. However, the SOM algorithm requires processing time which increases with the size of input data [49]. Hence, the DWT algorithm can be used to reduce the input data size in order to reduce transmission energy and eventually help prolong the overall network lifetime of the WSN without losing the significant data. The performances of the aforementioned algorithms were then evaluated by means of synthetic and real data injected with synthetic faults and also the real-world dataset. These experiments are shown in the next chapter.

CHAPTER III

EXPERIMENTS AND RESULTS

In this section, we evaluated the performance of the proposed integration of SOM and DWT algorithm by detecting anomalies in series of synthetic data and actual data collected from a wireless sensor network injected by various combinations of synthetic faults. We then conducted experiments to evaluate its performance by testing the proposed algorithm on real-world datasets with real-world faults obtained from various environmental, and microclimate sensor networks, and the dataset obtained from the WSN deployed at the SUT BOF plant.

3.1 Evaluation on detecting synthetic faults

In the experiment, we generated the synthetic input data from a normal distribution N(0,1) and synthetic faults by additive White Gaussian noise (AWGN) with power 25 dBW generated from MATLAB. We used such fault because its statistical similarity to the synthetic input data thus, it is more difficult to be detected. Therefore, we can evaluate the performance of the algorithms under ambiguous faults. The amount of faults was represented by the notation n/s, where “n” is the amount of faults per series and “s” is the amount of series of faults, resulting in the total amount of n×s faults. The generated faults added to the input data ranged from bursty which was 20/10, then 10/10, 2/10, and finally to sparse which was 1/10. The exact positions of the faults injected in the input data were predetermined and was later used to detect true and false alarms.

In the experiment using real data collected from wireless sensor nodes, we had chosen 2 parameters, namely soil temperature and soil moisture, as KPIs collected from samples of compost in the SUT bioorganic fertilizer plant. In this experiment, the data of the 2 KPIs at the WSNs were collected every 5 minutes for 3 days. We compared 3 anomaly detection methods: SOM algorithm, DWT algorithm, and integration of SOM and DWT algorithm.

We measured 2 performance metrics:

1) The *true alarm rate* which is defined by the number of detected true anomalies over the total number of true anomalies in the data set as shown below

$$\text{True alarm rate} = \frac{\text{Total detected true alarms}}{\text{Total true anomalies}} \times 100 \tag{3.1}$$

2) The *false alarm rate* which is defined by the number of detected false alarms over the total number of detected anomalies as shown below

$$\text{False alarm rate} = \frac{\text{Total detected false alarms}}{\text{Total normal data}} \times 100 \tag{3.2}$$

In the DWT algorithm, we used the threshold given by (3.3) and (3.4) in order to decide whether the data is normal or abnormal [50]

$$\sigma_w = median(\left|d_1 - \overline{d_1}\right|) \tag{3.3}$$

$$T_w = \sigma_w \sqrt{2\log_e(N)}\,, \tag{3.4}$$

where N is the size of data and $\overline{d}_1$ is the median of the level 1 detail coefficients.

This threshold was calculated from the low pass and high pass coefficients from the assumed normal data by using Haar and Daubechies4 mother wavelets. The Haar and Daubechies4 wavelets were used because they are relatively easy to cross-check by hand with computed coefficients from MATLAB program. Hence, we can compare the position of each coefficient with the actual fault position. After the threshold calculation, the set of coefficients which were obtained from the DWT of the noisy data were compared with the threshold, coefficient by coefficient. For the real data scenario, the data was normalized by equation (3.5) before being processed by the DWT to eliminate potential outliers:

$$\mathrm{Norm(Data)} = \frac{\mathrm{(Data) - mean(Data)}}{\sqrt{\mathrm{variance(Data)}}} \tag{3.5}$$

If the absolute value of the coefficient was greater than the computed threshold, an anomaly was said to be detected.

In the SOM algorithm and the proposed integrated SOM and DWT algorithm, the initial value for learning rate in the SOM part was set to $\eta_0 = 0.9$, and gradually reduced to $\eta_T = 10^{-5}$, in order to guarantee convergence. The number of training epochs was set to 50 because longer training epochs tend to over train the SOM Barreto et al. (2006). The required percentage of distance in (2.8) was set to 99%. We used a Gaussian neighborhood function because the distribution of the collected data after the normalization fits well to the Gaussian distribution.

Figures 3.2 and 3.3 demonstrated that the anomaly detection in SOM algorithm and the integrated SOM and DWT algorithms improved as the number of neurons was increased. This suggests that the more neurons used, the “finer” SOM’s classification becomes resulting in enhanced detection performance.

However, at neuron size 50×50, the SOM required much longer training time with a marginal improvement in the detection performance. Therefore, the 30×30 size of neurons was selected to train and test the SOM. We also improved the SOM algorithm by double checking with the univariate method in order to reduce the false alarm rate Barreto et al. (2006); Sukkhawatchani, P., and Usaha, W. (2008). To obtain accurate results, the performance metric was averaged over 70 runs, which gave the best accuracy as shown in Table 3.1.

Table 3.1 Accuracy results obtained by feeding synthetic input data to the 30×30 neuron SOM algorithm.

Runs	True alarm rate (%)	Deviation from previous runs
1	62.00	-
10	59.50	0.040
20	57.65	0.031
30	58.17	0.009
40	57.68	0.008
50	57.82	0.002
60	57.88	0.004
70	58.14	0.001
80	58.06	0.001
90	58.17	0.001
100	58.27	0.001

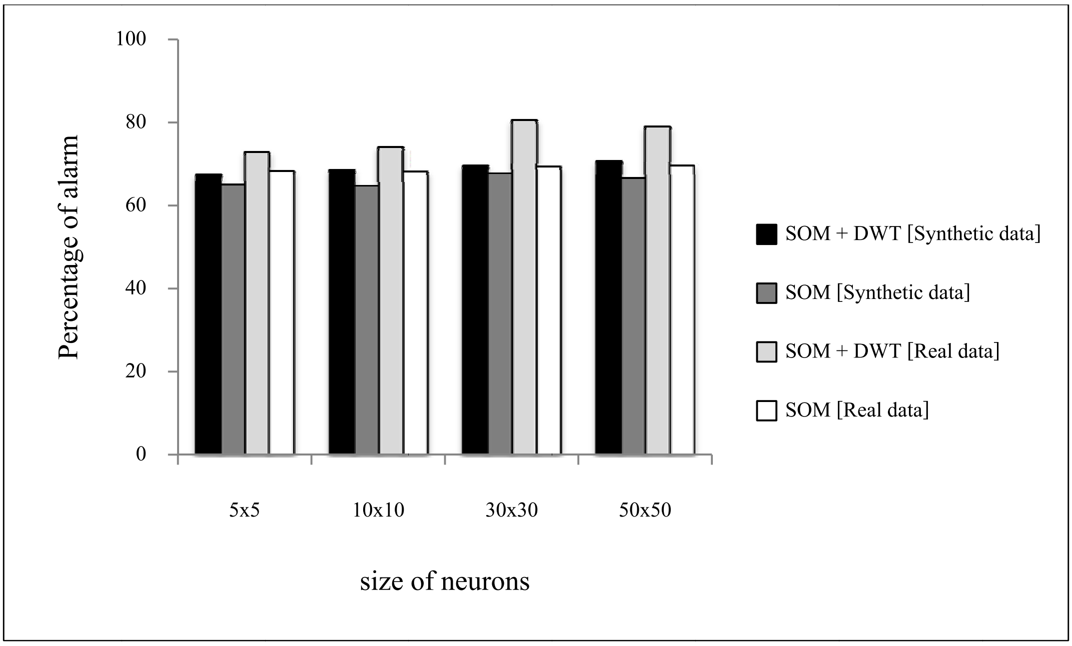


Figure 3.2 True alarm rates with different size of neurons in the sparse faults case.

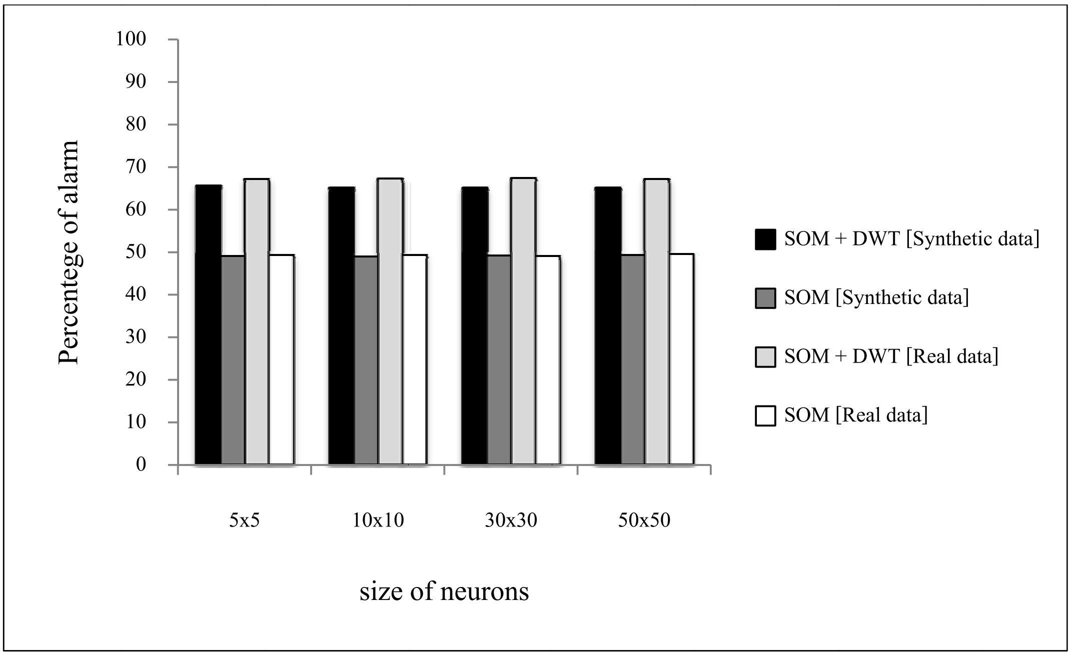


Figure 3.3 True alarm rates with different size of neurons in the bursty faults case.

To evaluate the performance of all algorithms, the results of each algorithm were compared to the (known) fault positions which were injected into the input data. In particular, when an anomaly was detected then its position was compared with the (known) fault position. If this position existed, then the anomaly detected was a true alarm; otherwise, it was a miss. On the other hand, if an anomaly was detected but the (known) fault position did not exist, then the anomaly was a false alarm.

Figures 3.4 and 3.5 showed the percentage of true alarm rate averaged over 70 runs, as a function of the amount of faults added into the input data. Note that the proposed integrated SOM and DWT algorithm which used Haar as a mother wavelet gave the best performance over other algorithms. This is because the DWT with Haar wavelet can detect changing points. In particular, the Haar wavelet uses 2 adjacent input data to compute a coefficient whereas the Daubechies4 uses 4 adjacent input data to compute a coefficient. However, Daubechies4 gave a lower performance than Haar because each coefficient was computed from an average over 4 input data. If a fault occurred in 1 of these 4 data, such fault would be averaged with the remaining 3 normal data resulting in a coefficient with an absolute value possibly lower than the decision threshold. Consequently, the true alarm rate was reduced. On the other hand, the Haar wavelet only used 2 adjacent data to compute 1 coefficient. Thus, the true alarm rate was significantly higher than that of Daubechies4 (db4). The integrated SOM and DWT algorithm using Haar also outperformed the SOM algorithm. This is because in the Haar case, the coefficients obtained were transformed from two adjacent data. Therefore, if some data were faulty or differed greatly from the data nearby, this coefficient can detect such anomaly. On the other hand, the SOM algorithm directly checked the data one by one to detect an anomaly. If the data were faulty but had a small magnitude, then this fault may not be detected, and consequently the true alarm rate was reduced. Note that the DWT algorithm had the lowest performance because the decision threshold in (3.4) is rather conservative. Furthermore, the threshold was changed throughout the detection and the algorithm did not have any double checking method.

In Figures 3.4 and 3.5, the proposed algorithm can achieve up to 65% and 67% of true alarm rates in case of bursty faults for synthetic and real data, respectively. The proposed algorithm achieved a true alarm rate of up to 18% higher than the SOM algorithm alone in presence of bursty faults. Compared to the DWT alone, the proposed algorithm can attain a true alarm rate of up to 35% more in the bursty faults case.

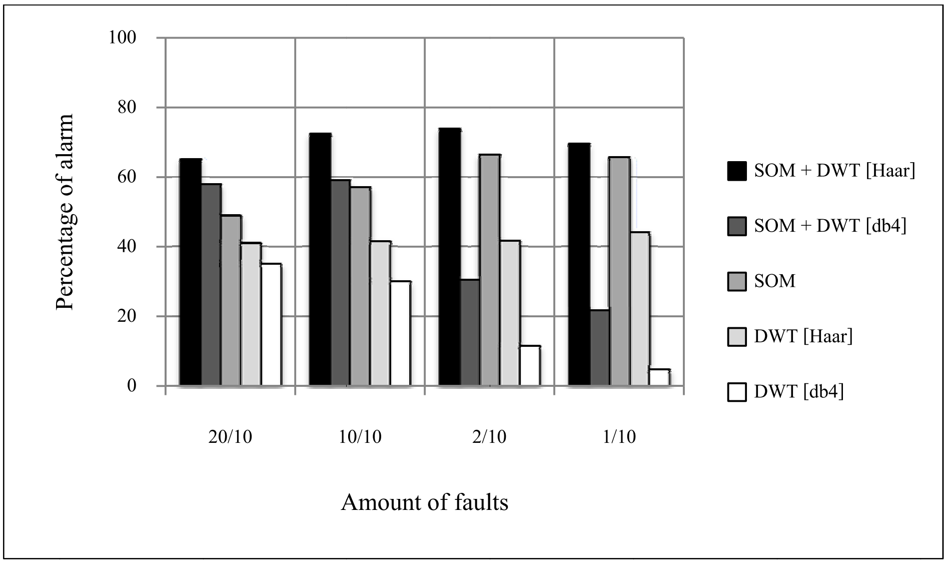


Figure 3.4 True alarm rates with synthetic data.

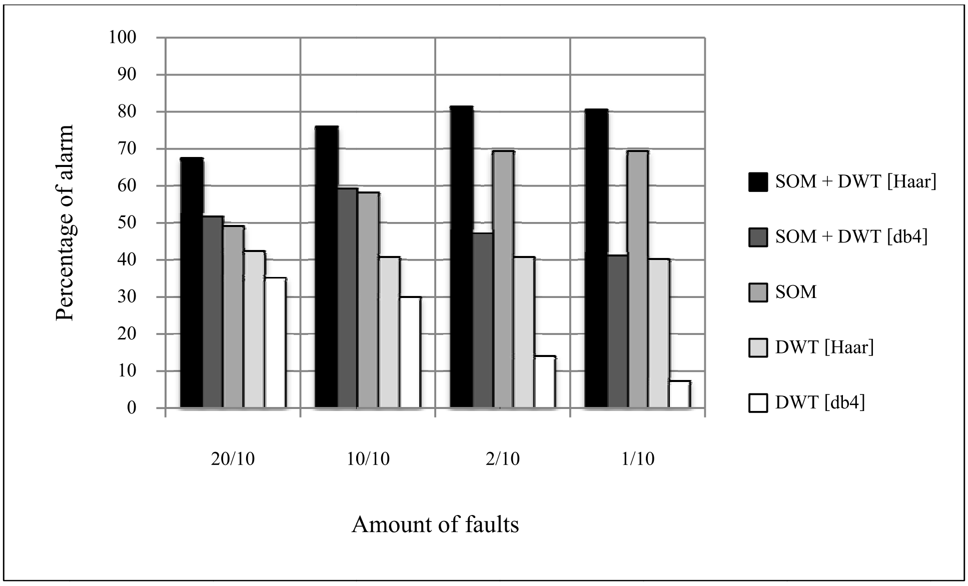


Figure 3.5 True alarm rates with real data.

As for sparse faults, the proposed algorithm can achieve up to 69% and 80% true alarm rates for synthetic and real data, respectively. The integrated SOM and DWT also gave true alarm rates of up to 10% higher than the SOM algorithm alone whereas DWT performed the weakest, in presence of sparse faults.

Figures 3.6 and 3.7 depicted the false alarm rate results in the synthetic and real data experiments, respectively. Note that most results had low false alarm rates, i.e., less than 1 % except in the case of sparse faults due to the increased detection difficulty.

The integration of SOM and Daubechies4 DWT also gave a weak performance due to the reasons previously explained. All these results showed that the integration of SOM and DWT with Haar as a mother wavelet outperformed the SOM algorithm and DWT method.

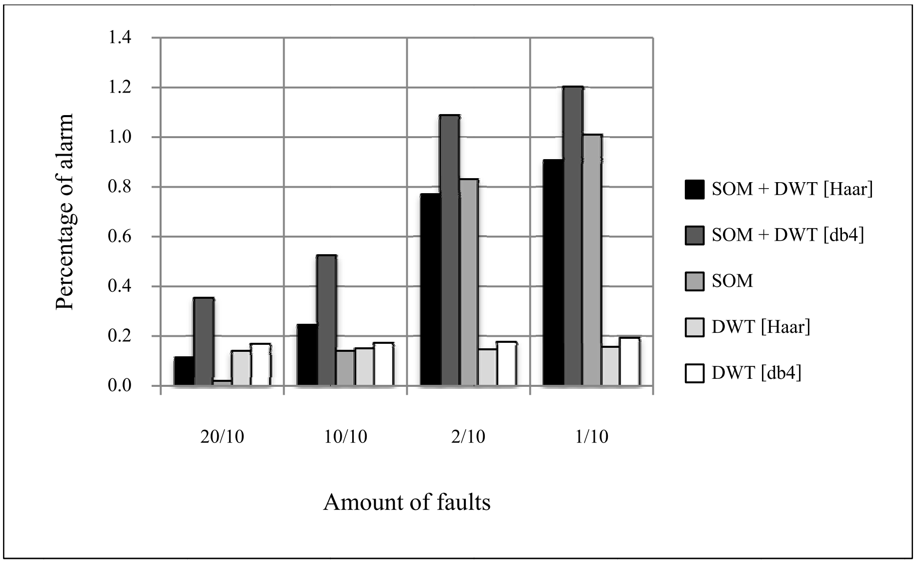


Figure 3.6 False alarm rates with synthetic data.

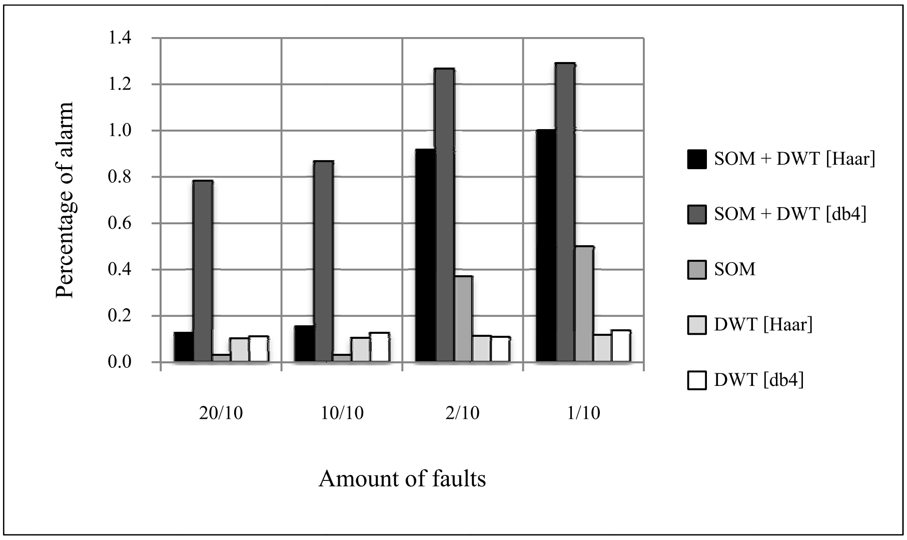


Figure 3.7 False alarm rates with real data.

From these figures, the false alarm rate of the proposed algorithm was 0.11% and 0.13% in presence of bursty faults and 0.91% and 1% in presence of sparse faults with synthetic and real data, respectively. Note that the false alarm rate of the proposed algorithm was slightly higher than the other two algorithms.

Since the gain in the true alarm rate was more significant, such tradeoff was therefore considered acceptable.

Figures 3.8 and 3.9 illustrated the effect of the decreasing of AWGN noise power from 25dBW to 10dBW in both synthetic and real data scenarios. Only the Haar wavelet was used in the proposed algorithm and the DWT algorithm. The Daubechies4 was not included due to its weak performance. Though the anomaly detection was more difficult, the proposed integrated SOM and DWT still consistently outperformed the other two methods in terms of true alarm rate but with marginal increased in the false alarm rate as tradeoff.

The proposed integration of SOM and DWT algorithm with Haar wavelet outperformed the SOM algorithm and the DWT algorithm alone. Our results demonstrated that the proposed integrated SOM and DWT anomaly detection scheme can be deployed in a resource-constrained network such as a WSN. In particular, the DWT using Haar wavelet can be implemented at the sensor nodes as a data preprocessor to reduce the amount of data to be transmitted by at least half (for one level DWT). Since energy consumption is critical in WSNs, such distributed in network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN (Rajasegarar, Leckie, and Palaniswami, 2008). while still maintaining acceptable anomaly detection accuracy.

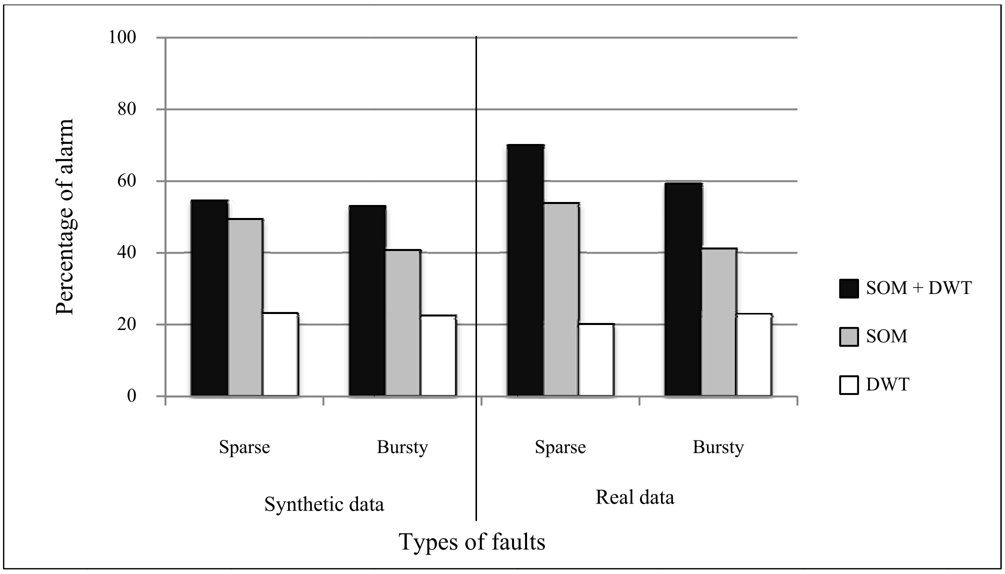


Figure 3.8 True alarm rates with 10 dBW AWGN faults.

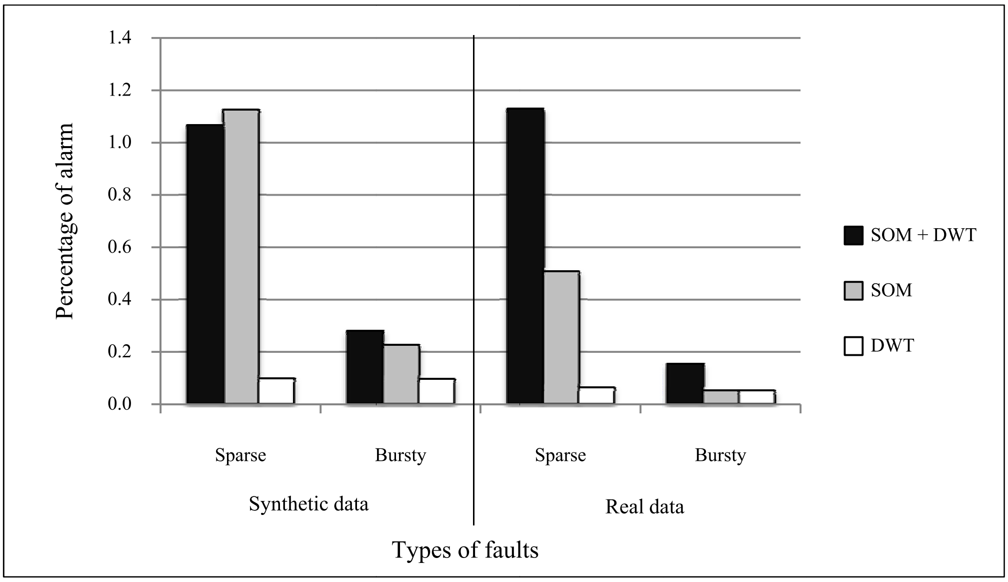


Figure 3.9 False alarm rates with 10 dBW AWGN faults.

3.2 Evaluation on detecting faults in real-world datasets

In this section, we applied the anomaly detection methods to three real-world datasets, i.e., (NAMOS, 2006) INTEL Berkeley Lab (INTEL, 2004); SensorScope (2006) to detect anomalies in sensor traces. However, since we did not have ground truth information about faults for these datasets, visual inspection and the histogram method were used to decide whether the data is normal or abnormal. The histogram method was used because it displays the data distribution which allows us to determine a suitable threshold according to that data series.

In the histogram method, we divided the time series of sensor readings into groups of *N* samples. We then plotted the histogram of the samples and selected a threshold according to outliers of the histogram. However, this approach was sensitive to the choice of *N*. Figure 3.10 showed the effect of *N* on the histogram computed for sensor measurements taken from a real-world deployment (Sharma, Golubchik, and Govindan, 2010). Therefore, the selection of the correct value for the parameter *N* required a good understanding of the *normal* sensor readings. In practice, one should also try a range of values for *N* to ensure that the samples flagged as faulty are not just a result of the value selected for *N* (Sharma, Golubchik, and Govindan, 2010). With heuristic adjustments on the parameter value of N and some domain knowledge of the normal data profile, the histogram method was used as reference to identify abnormal data samples.

In the real-world datasets experiment, we evaluated the performance of 3 anomaly detection methods: the SOM, DWT using the Haar wavelet methods, and the integration of SOM and DWT using the Haar wavelet. We did not consider DWT using the db4 wavelet because it obtained poor detection performance as shown in the synthetics faults experiment. For the SOM and the integration of SOM and DWT using Haar wavelet algorithms, we also considered the effects of changing the number of training samples, the number of training epochs which were 10 and 50 iterations, and the size of neurons which were 10x10 and 30x30.

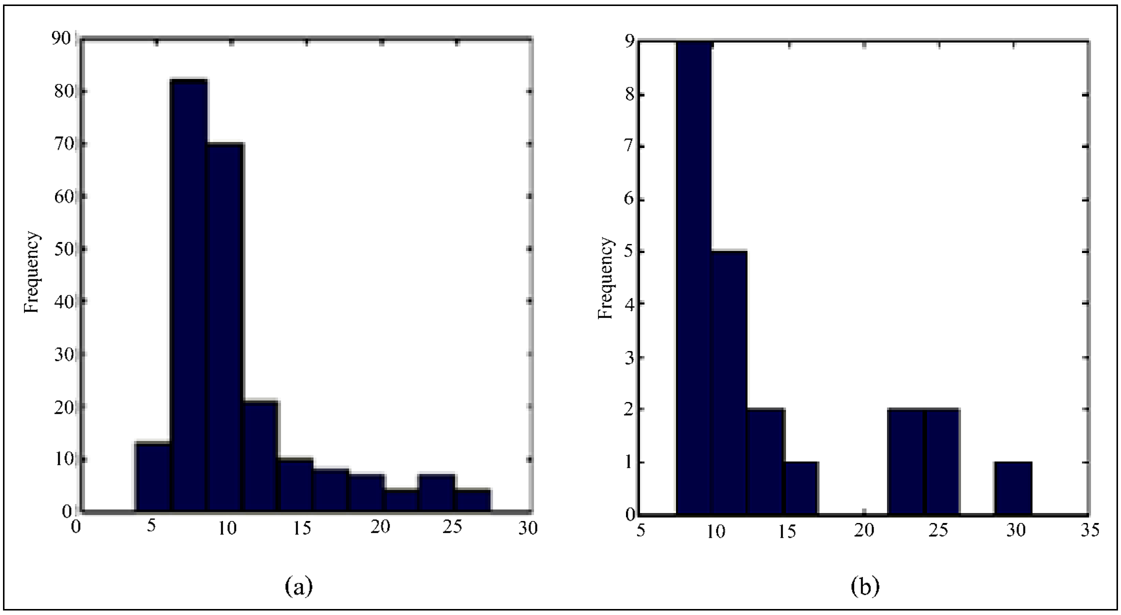


Figure 3.10 Histogram shape with N = 100 (a) and N = 1000 (b)

We also compared the performance of the low and high pass Haar wavelet coefficients (LP and HP, respectively) in the DWT algorithm and the integration of SOM and DWT algorithm. This is because each coefficient can perform well for different types of faults so we can choose the suitable coefficients for the data.

3.2.1 NAMOS

In the NAMOS dataset, 9 buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, for over 24 hours in August, 2006 (NAMOS,2006). We analyzed the measurements from chlorophyll sensors on buoys no. 103 for 10^4 samples as shown in Figure 3.11. In the experiment, the histogram method was used to identify anomalies in the NAMOS dataset from which we selected the threshold of 0 and 500 as lower and upper bounds of the normal region, respectively. The sizes of training samples of 1500 and 3000 samples were used to train both the SOM and the integration of SOM and DWT algorithms.

Figure 3.12 showed the percentage of detection alarm rates for true, miss and false alarms which were obtained from changing the sizes of training samples. Note that both the SOM algorithm and the proposed integrated SOM and DWT algorithm with low pass wavelet coefficients gave the best true alarm detection performance of nearly 100% while their false alarm rates were negligible. The integrated SOM and DWT algorithm and DWT algorithm with high pass coefficients gave the lowest performance. This is because the high pass coefficients are more suitable for detecting the changing points of the data whereas most of faults appear constant as seen from $9x10^3$ samples onwards in Figure 3.11. In addition, reducing the size of training samples did not have any effect on the anomaly detection in the SOM algorithm and the proposed integrated SOM and DWT algorithm. This was because both training samples were obtained from a normal period of data which differ only in sample sizes.

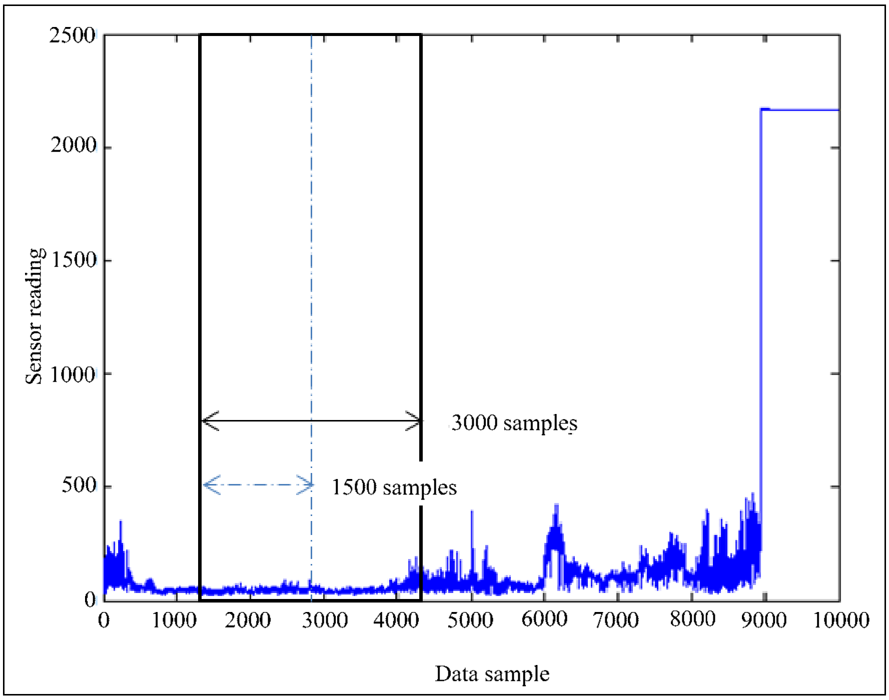


Figure 3.11 NAMOS dataset of 10^4 samples.

Figure 3.13 depicted the percentage of detection alarm rates for true, miss, and false alarms which were obtained by reducing the number of training epoch from 50 to 10 iterations. In this case, the SOM algorithm gave the best performance with nearly 100% of true alarm detection rate and no false alarm rate. DWT algorithm which used low pass coefficient gave high performance while the proposed integrated SOM and DWT algorithm with either coefficient failed on detecting any anomaly. The reason could be caused by the constant features of the faults in NAMOS which may be difficult to decide whether samples are normal or abnormal, in particular, if the wavelet coefficients were under-trained. Hence, care must be taken when selecting the suitable number of training epochs. In addition, we also investigated the effect of reducing the size of neurons. Results in Figure 3.14 showed that there was no significant effect from reducing size of neurons from 30x30 to 10x10.

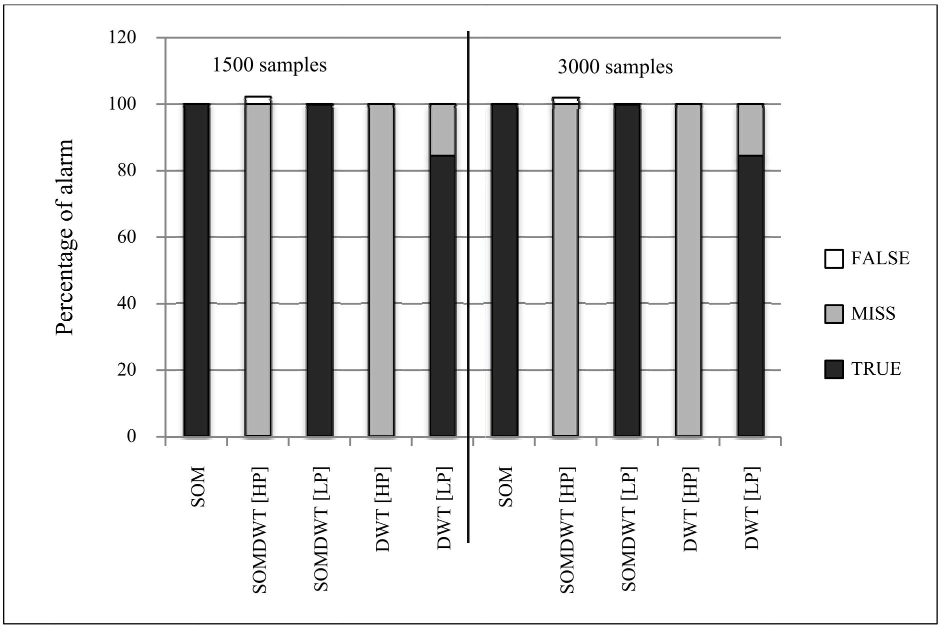


Figure 3.12 Detection rate in the NAMOS dataset using training epoch of 50 iterations.

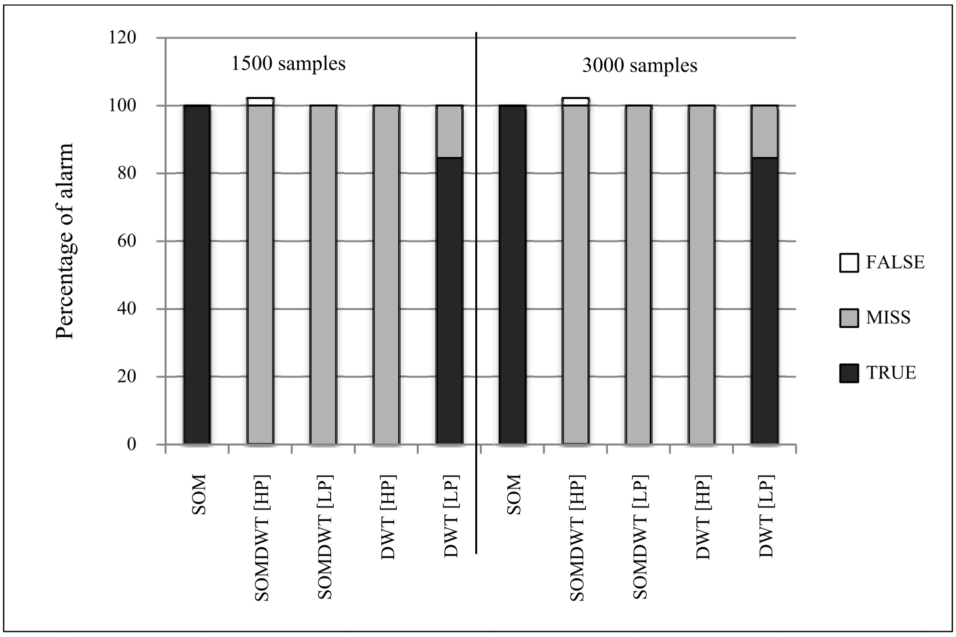


Figure 3.13 Detection rate in the NAMOS data set using training epoch of 10 iterations.

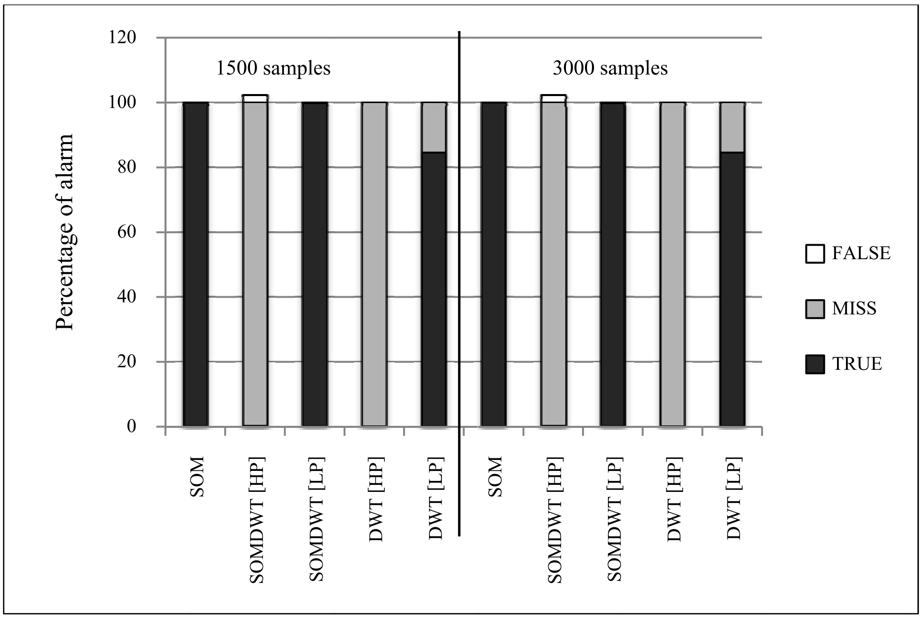


Figure 3.14 Detection rate in the NAMOS dataset using the size of neurons of 10x10

3.2.2 INTEL

In the INTEL dataset, 54 Mica2Dot motes with temperature, humidity and light sensors were deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004 (INTEL, 2004). For this experiment, we presented the results on the anomaly detection in the temperature readings.

We selected the threshold value of 16 and 30 as the upper and lower bounds of the normal data regions. These values were obtained from the histogram method. The sizes of training samples used were 1000 and 2000 samples as shown in Figure 3.15.

Figure 3.16 showed the percentage of detection alarm rates for true, miss, and false alarms which were obtained from changing the size of training samples. According to the results as shown, the SOM and the proposed algorithm can achieve a true alarm rate of up to 100% with very small false alarm rate. Their true alarm rate was 67% higher than the DWT method using high pass coefficients. Note that the high pass coefficients can detect spike faults better than low pass coefficient since the high pass coefficients reflect the rate of change between two successive samples. Note that the DWT using low pass coefficient gave the lowest performance. The results of changing number of training epochs were shown in Figure 3.17 and the sizes of neurons were shown in Figure 3.18. From both figures there were no significant effects on the detection rate because the fault in this dataset had high amplitude and can be easily detected.

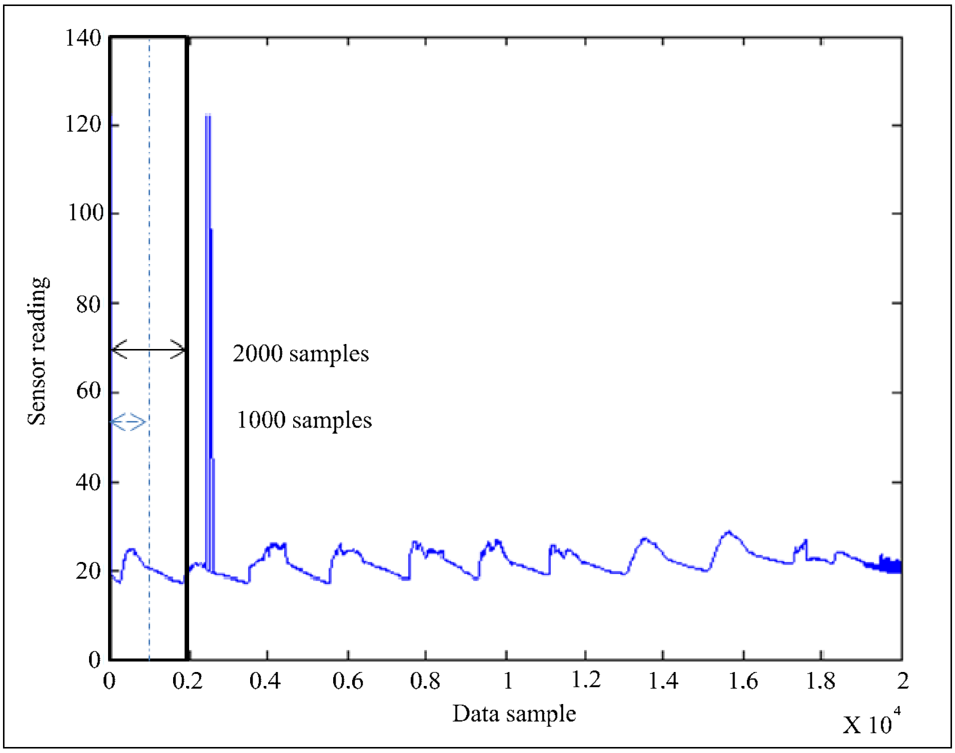


Figure 3.15 INTEL dataset of 2×10^4 samples.

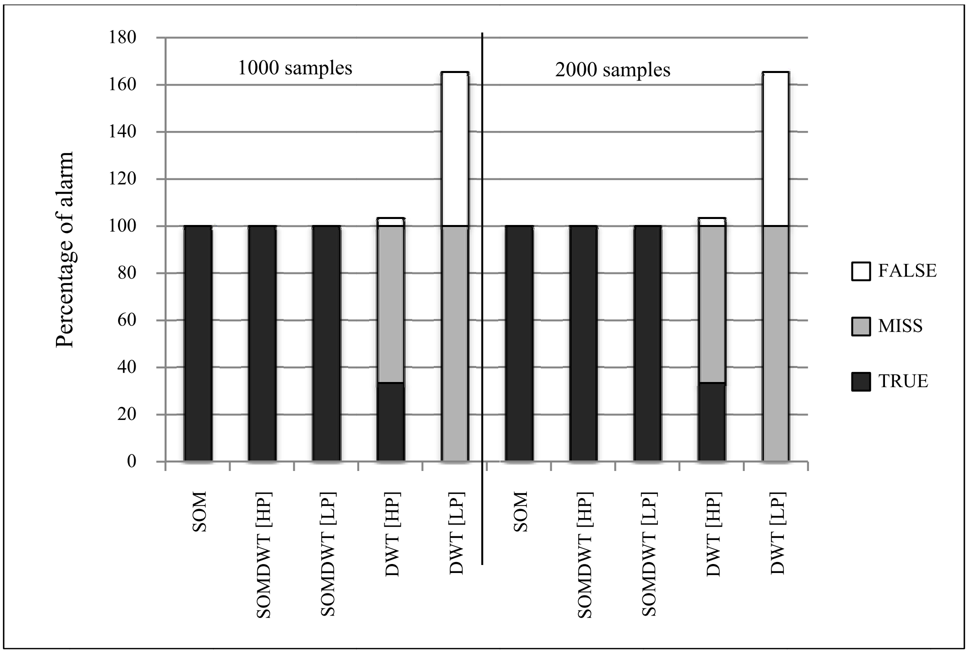


Figure 3.16 Detection rate in the INTEL dataset using training epoch of 50 iterations.

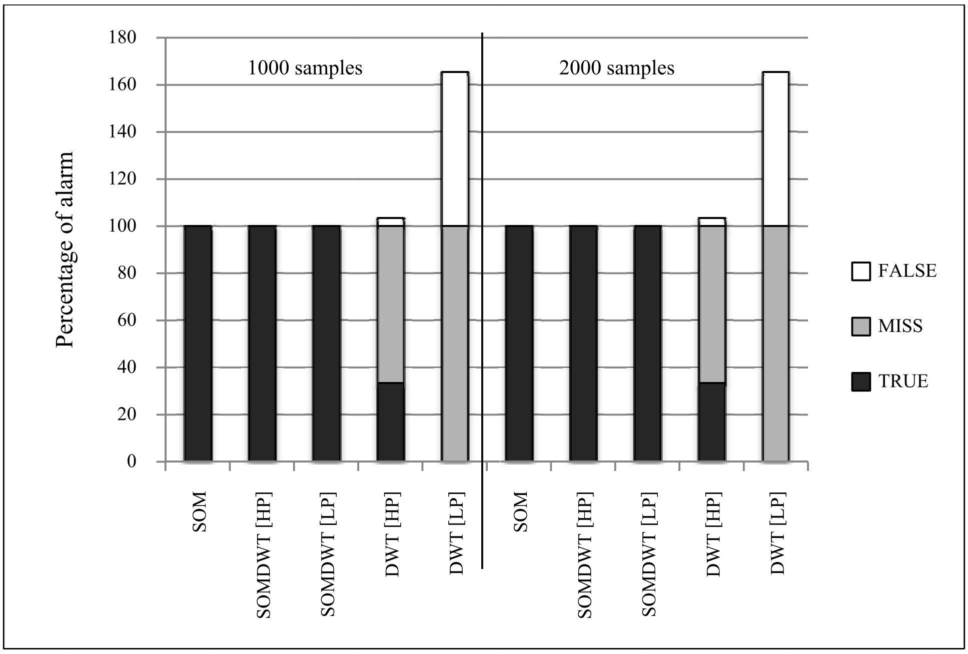


Figure 3.17 Detection rate in the INTEL dataset using training epoch of 10 iterations.

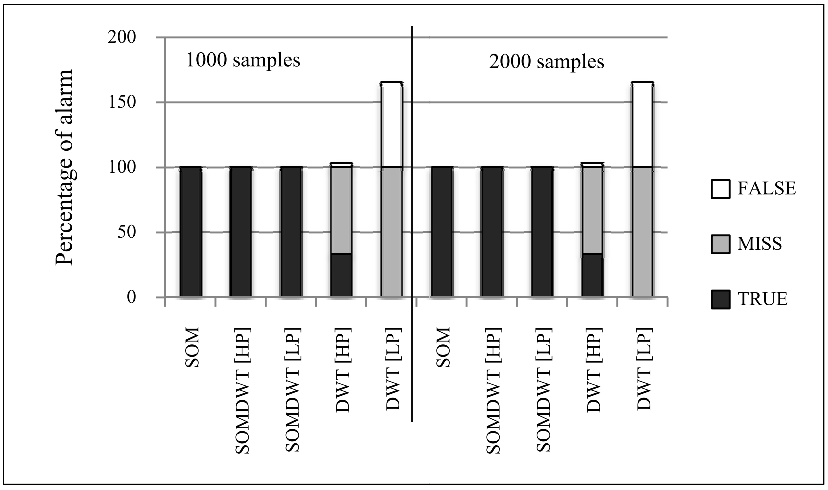


Figure 3.18 Detection rate in the INTEL dataset using the size of neurons of 10x10.

3.2.3 SensorScope

The SensorScope project was an ongoing outdoor sensor network deployment consisting of weather-stations with sensors for sensing several environmental quantities such as temperature, humidity, solar radiation, soil moisture etc. (SENSORSCOPE, 2006) We did not have the ground truth regarding faulty samples for this dataset. We used a combination of visual inspection and the histogram method to identify anomaly samples (Sharma, Golubchik, and Govindan, (2010).

3.2.3.1 SensorScope station no.39 dataset

In this experiment, we presented the results on anomaly detection in one KPI of SensorScope which was collected from weather station no.39 (SensorScope39). Using visual inspection and the histogram method, the lower and upper threshold valued used for anomaly detection in SensorScope were 1.5 and 9. The sizes of training samples were 1600 and 3200 samples as illustrated in Figure 3.19.

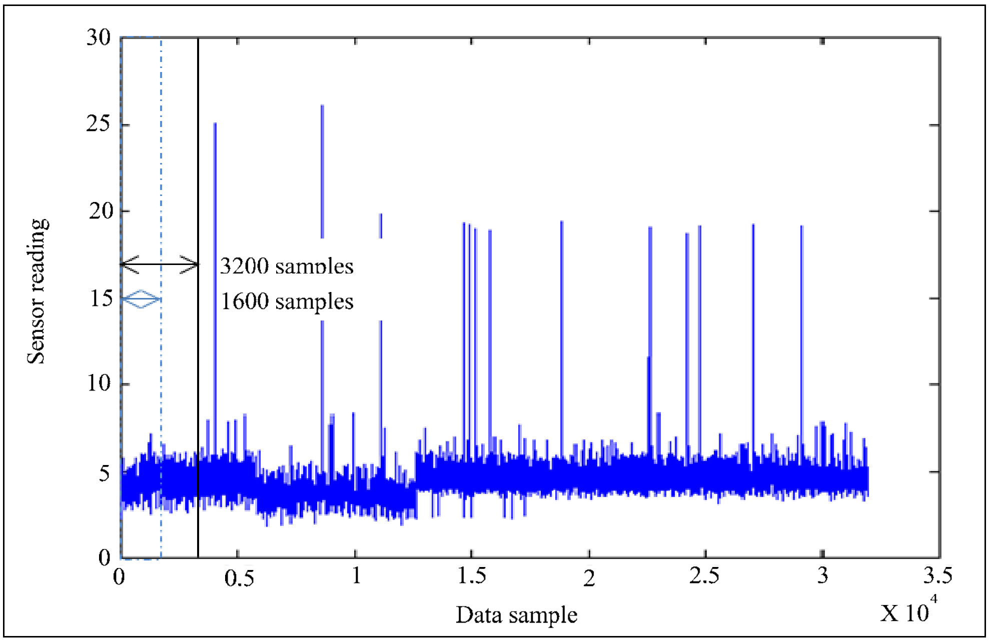


Figure 3.19 SensorScope39 dataset of 32000 samples.

Figure 3.20 depicted the percentage of detection alarm rates which were obtained from changing the size of training samples. According to the results as shown, the SOM and the proposed algorithm can achieve a true alarm rate of up to 100% with very small false alarm rate. The DWT method using high pass coefficients also gave true alarm rate of up to 100% but gave high false alarm rate. Their true alarm rate was 8% higher than the DWT method using low pass coefficients. Note that the high pass coefficients can detect spike faults better than low pass coefficients since the high pass coefficients reflect the rate of change between two successive samples. The results of changing number of training epochs were shown in Figure 3.21 and the sizes of neurons were shown in Figure 3.22. From both figures, there were no significant effects on the detection rate because the fault in this dataset has a high amplitude and can be easily detected.

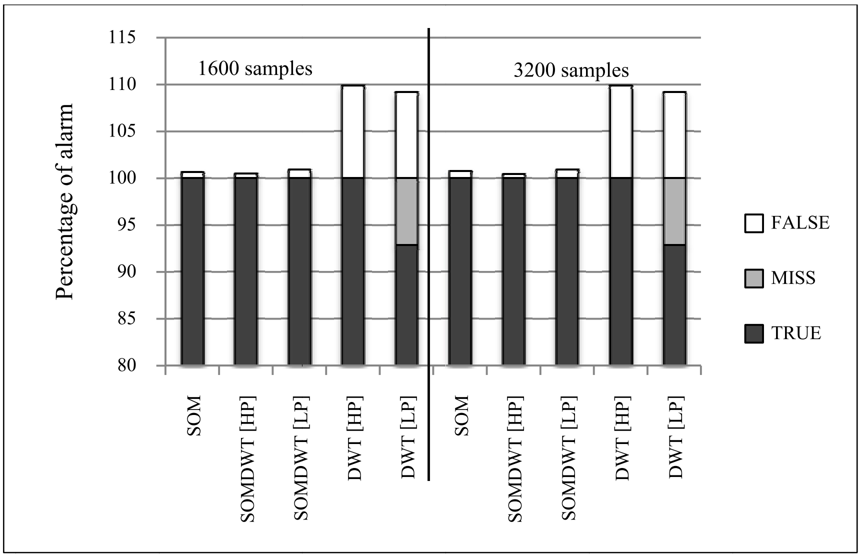


Figure 3.20 Detection rate in the SensorScope39 of first KPI dataset using training epoch of 50 iterations.

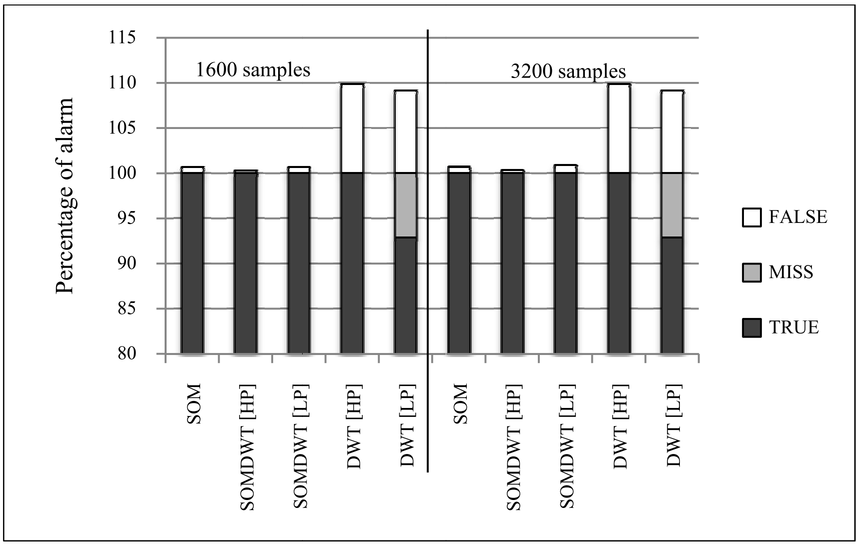


Figure 3.21 Detection rate in the SensorScope39 of first KPI data set using training epoch of 10 iterations.

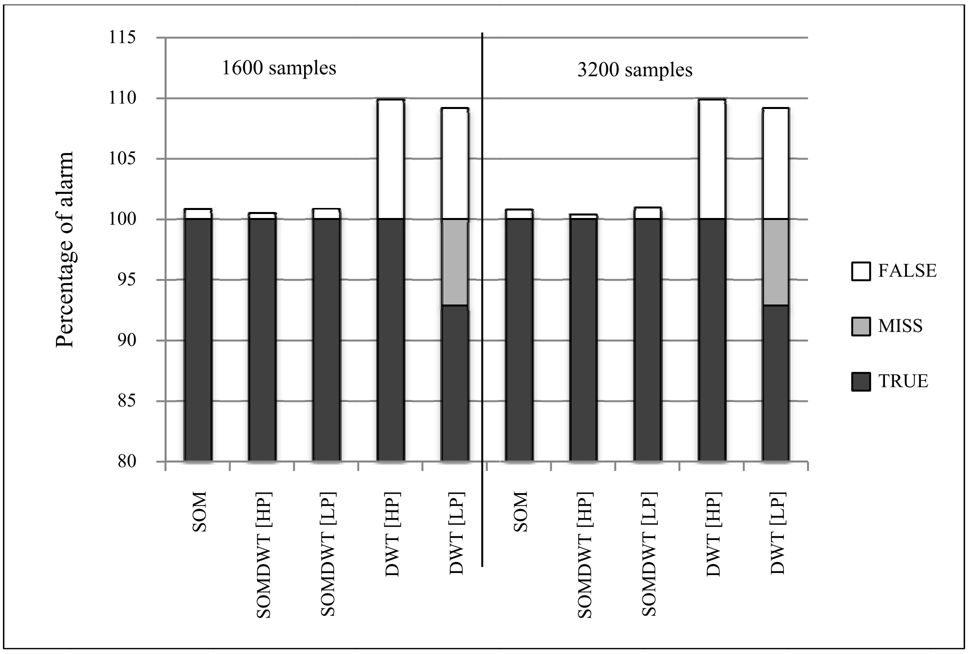


Figure 3.22 Detection rate in the SensorScope39 of first KPI dataset using the size of neurons of 10x10.

3.2.3.2 pdg2008-metro-1 dataset

In the experiment, we presented the results on the anomaly detection in two types (KPIs) of data in the pdg2008-metro-1 dataset, i.e., the surface and ambient temperature readings. Using visual inspection and the histogram method, the lower and upper threshold values used for anomaly detection in SensorScope were -14 and 4 for the surface temperature and -12 and 4 for the ambient temperature. The sizes of training samples were 700 and 2000 samples for both KPIs as shown in Figure 3.23.

Figure 3.24 showed the percentage of detection alarm rates for true, miss, and false alarms obtained from changing the size of training samples. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 2% higher than the SOM algorithm while false alarm rate remained less than 0.5%. The proposed algorithm using low pass coefficients can attain a true alarm rate of up to 17% more than the DWT algorithm alone. The integrated SOM and DWT algorithm and DWT algorithm which used high pass coefficients gave the lowest performance. This is because high pass coefficients were more suitable for short duration faults such as spike or sparse faults while the data in Figure 3.23 contained noise faults which affected a larger number of successive samples with an increase in their variance. The effect of reducing the number of training epochs was shown in Figure 3.25. According to the results, there was no significant effect on the performance of SOM and the integrated SOM and DWT. In Figure 3.26, the percentage of detection alarm rates for true, miss, and false alarms were obtained from reducing the size of neurons. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 2% lower than the SOM algorithm, whereas the false alarm rate remains lower than 0.5%. On the other hand, the proposed algorithm using low pass coefficients can attain a true alarm rate of up to 13% more than the DWT algorithm alone.

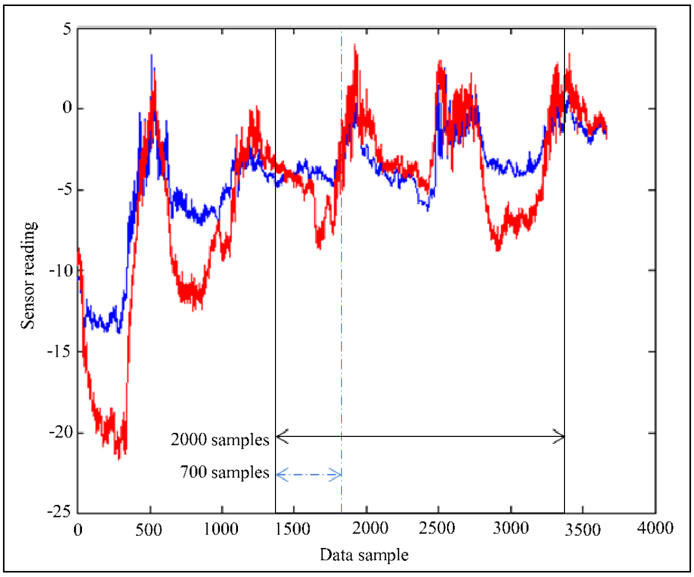


Figure 3.23 SensorScope pdg dataset of 4000 samples.

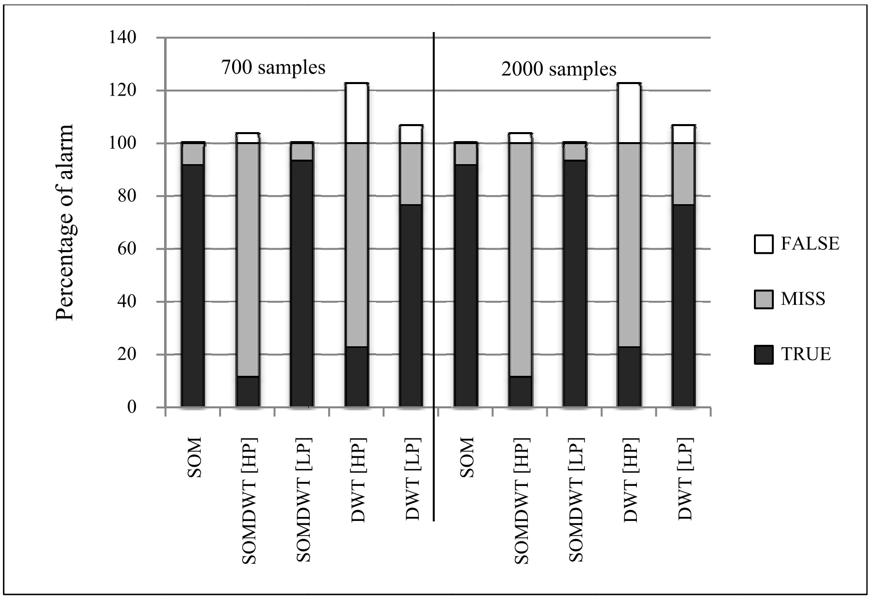


Figure 3.24 Detection rate in the SensorScope pdg dataset using training epoch of 50iterations.

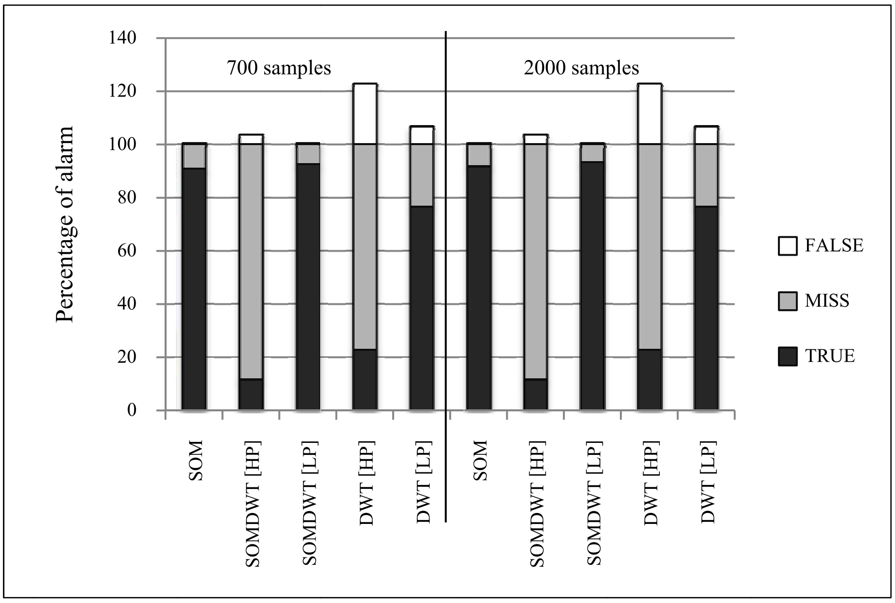


Figure 3.25 Detection rate in the SensorScope pdg dataset using training epoch of 10 iterations.

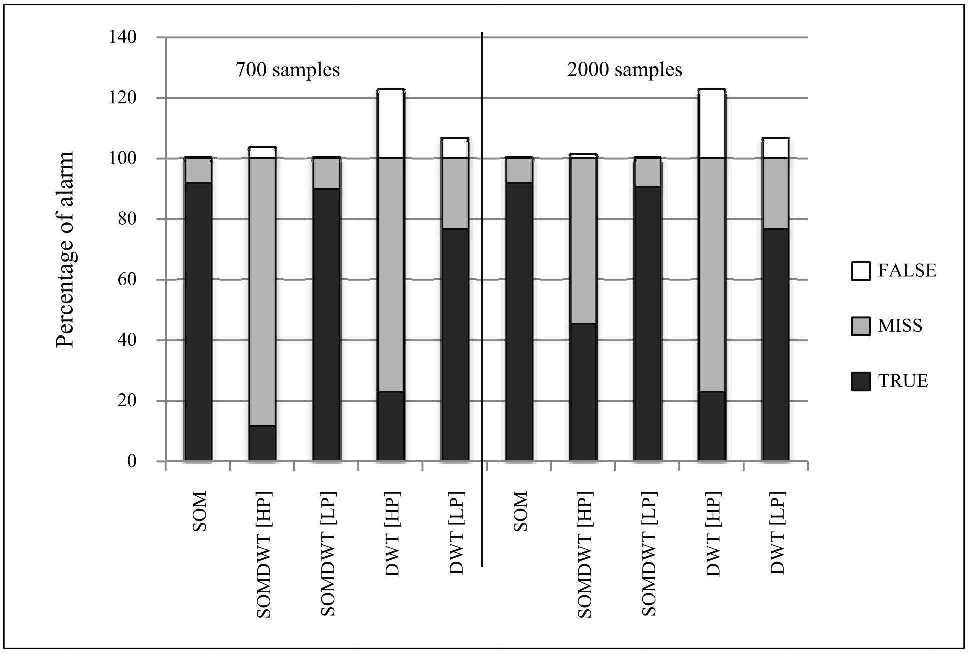


Figure 3.26 Detection rate in the SensorScope pdg dataset using the size of neurons of 10x10.

3.3 Evaluation on detecting faults in bioorganic fertilizer datasets

In this research, we also applied the anomaly detection algorithms to a BOF dataset which was collected from the prototype of WSN deployed at SUT BOF plant.

To collect data from the prototype, we have designed a system suitable for extreme conditions in the fertilizer compost. The preliminary design for the prototype system consists of base station, sensor mote, and sensor probes for soil moisture and soil temperature which were mounted onto a post. A number of such posts were installed at two locations within the compost. The motes monitor and transmit data continuously to the base station. The design allowed the posts to be easily removed before the compost is turned over.

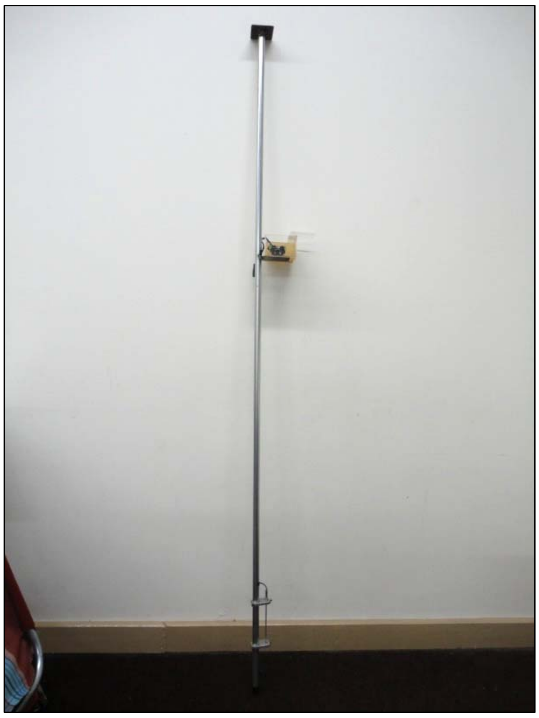


Figure 3.27 Prototype post.

At the gateway or base station, we used Crossbow's MIB520CB (USB-interface board). The MIB520CB provides USB connectivity to the IRIS or MICA family of motes for communication and in-system programming. Any IRIS/MICAz/MICA2 node can function as a base station when connected to the MIB520CB USB interface board.

In addition to data transfer, the MIB520CB also provides a USB programming interface. The MIB520CB offers two separate ports: one dedicated to in-system Mote programming and the second for data communication over USB interface. The MIB520CB has an on-board processor that programs Mote Processor Radio Boards. USB Bus power eliminates the need for an external power source.

At the sensor mote, we used Crossbow's MPR2400 (MicaZ). The MPR2400 is based on the Atmel ATmega128L which is a low-power microcontroller that runs

MoteWorks from its internal flash memory. A single processor board (MPR2400) can be configured to run our sensor application/ processing and the network/radio communications stack simultaneously. The 51-pin expansion connector supports Analog Inputs, Digital I/O, I2C, SPI, and UART interfaces. These interfaces make it easy to connect to a wide variety of external peripherals.

For the data acquisition board, we used Crossbow's MDA300. The data Acquisition board (DAQ) is used to get information from variety of different sources. These sources can be a hardware DAQ attached to the local or remote running machine, a remote TCP or UDP connection or any different sensors. DAQ software provides interface of the data sources to different outputs. We used two sensor devices which are soil temperature sensor and soil moisture sensor. Thermocouples are widely used temperature sensors which can also be used to convert heat into electric power. They are cheap and interchangeable, have standard connectors, and can measure a wide range of temperatures. As for the soil moisture sensor probe, we used EC-5 because it can tolerate high temperatures within the pile of compost.

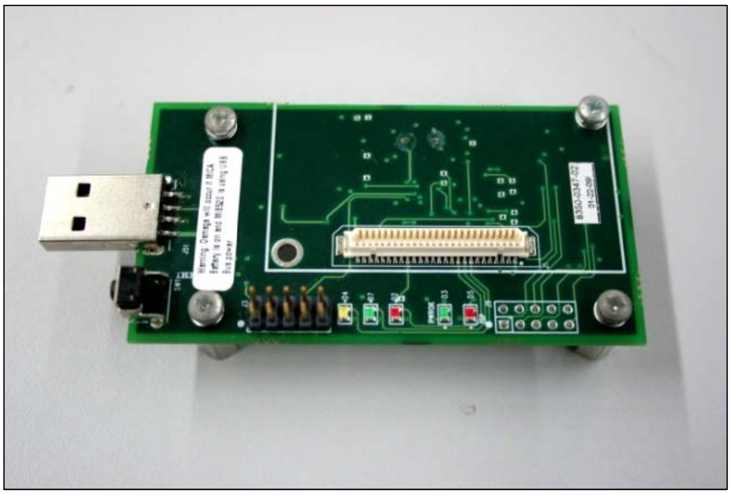


Figure 3.28 MIB520CB (USB interface board).

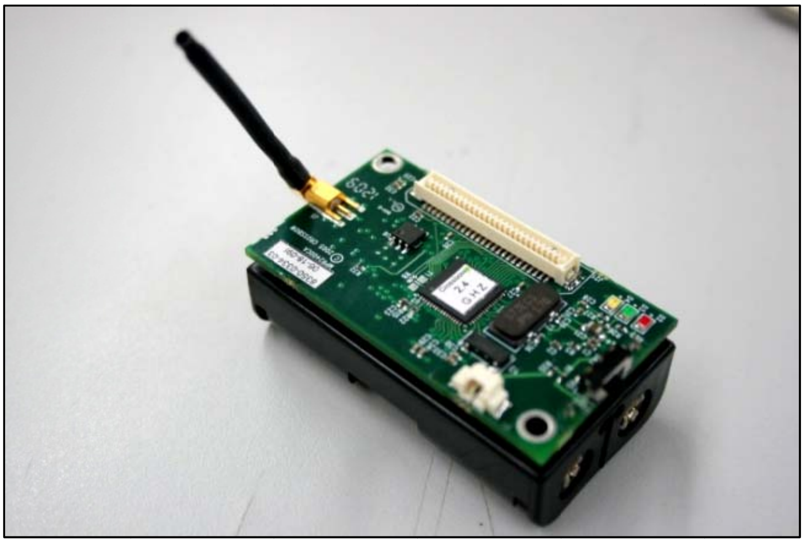


Figure 3.29 Sensor node model MPR2400 or MicaZ.

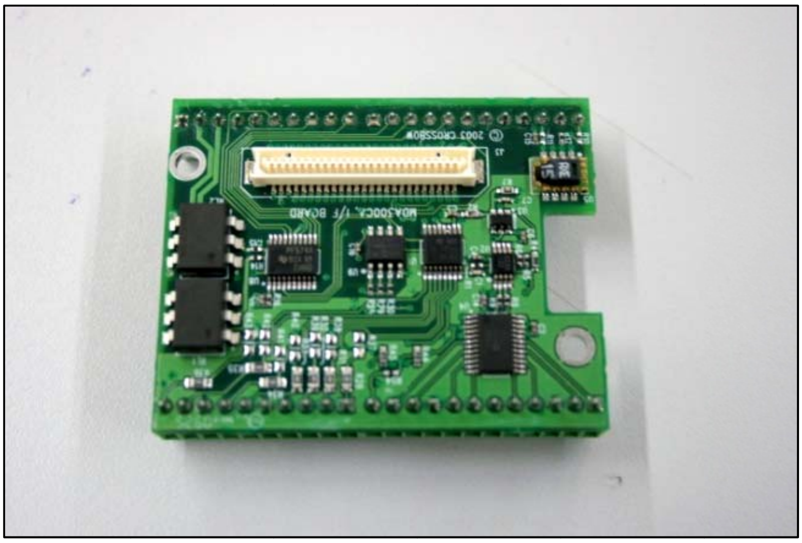


Figure 3.30 Data acquisition board model MDA300.

Figure 3.31 Temperature sensor (Thermocouple).

Figure 3.32 Moisture sensor (EC-5).

We performed anomaly detection on one type (KPI) of data in the BOF dataset, i.e., the temperature readings which was collected every 5 minutes for a day. Using visual inspection and the histogram method, the lower and upper threshold values used for anomaly detection in BOF dataset were 0 and 30. The sizes of training samples were 1000 and 2000 samples as shown in Figure 3.33.

Figure 3.34 depicted the percentage of detected alarm rates for true, miss, and false alarms obtained from changing the size of training samples. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 5% less than the SOM algorithm while false alarm rate remained lower than 3%. The proposed algorithm using low pass coefficients can attain a true alarm rate of up to 75% more than the DWT algorithm alone. The integrated SOM and DWT algorithm and DWT algorithm which used high pass coefficients gave low performance. This was because high pass coefficients were more suitable for short duration faults such as spike or sparse faults while the data in Figure 3.33 contained noise faults which affected a larger number of successive samples with an increase in their variance.

The effect of reducing the number of training epochs was shown in Figure 3.35. According to the results, there was no significant effect on the performance of SOM and the integrated SOM and DWT. Similarly, in Figure 3.36, no significant changes were found in the percentage of detected alarm rates for true, miss, and false alarms by reducing the size of neurons.

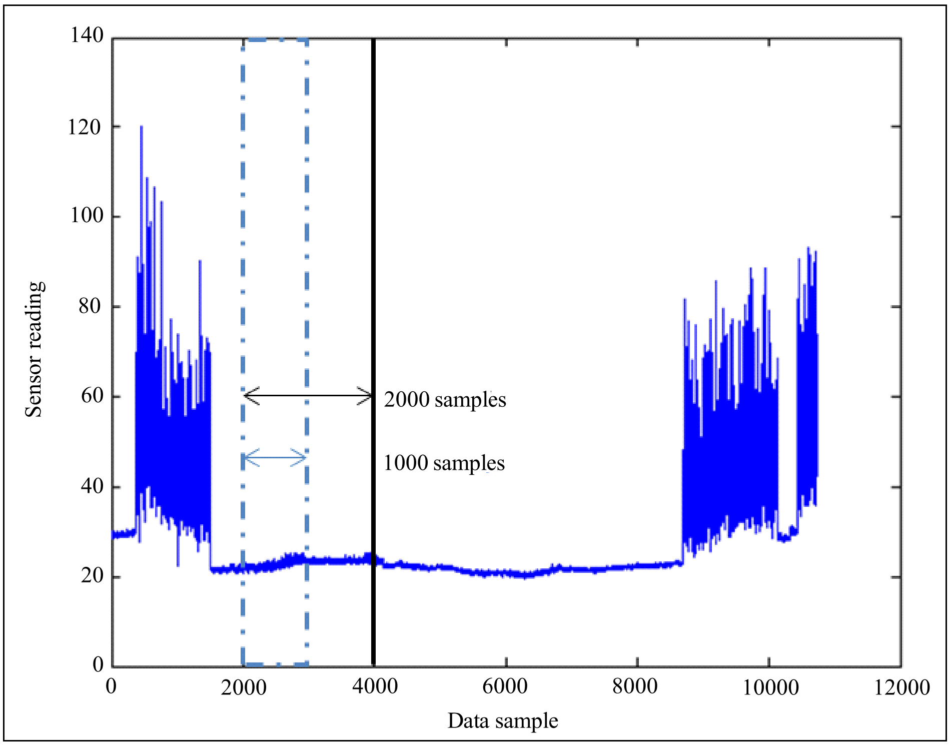


Figure 3.33 Bioorganic fertilizer dataset of 10740 samples.

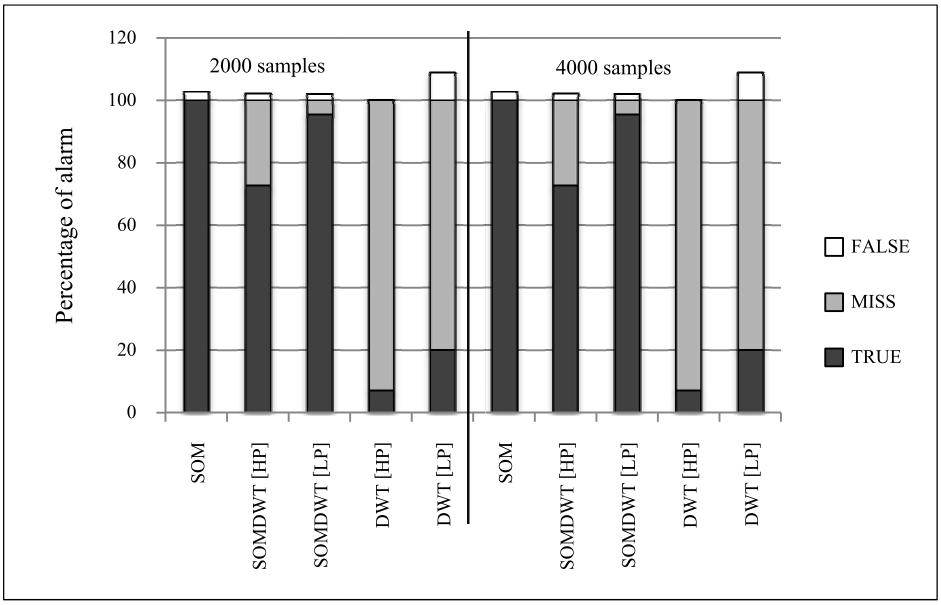


Figure 3.34 Detection rate in the bioorganic fertilizer dataset using training epoch of 50 iterations.

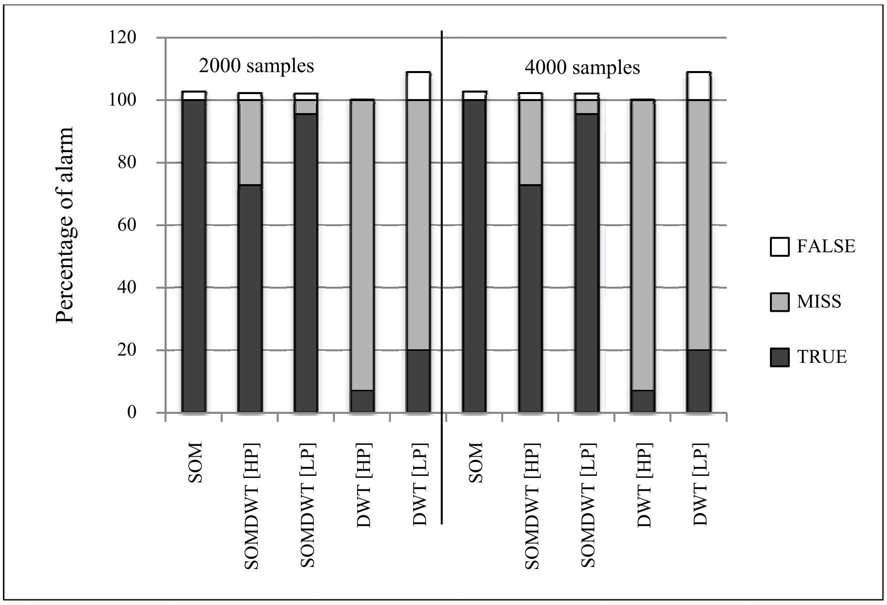


Figure 3.35 Detection rate in the bioorganic fertilizer dataset using training epoch of 10 iterations.

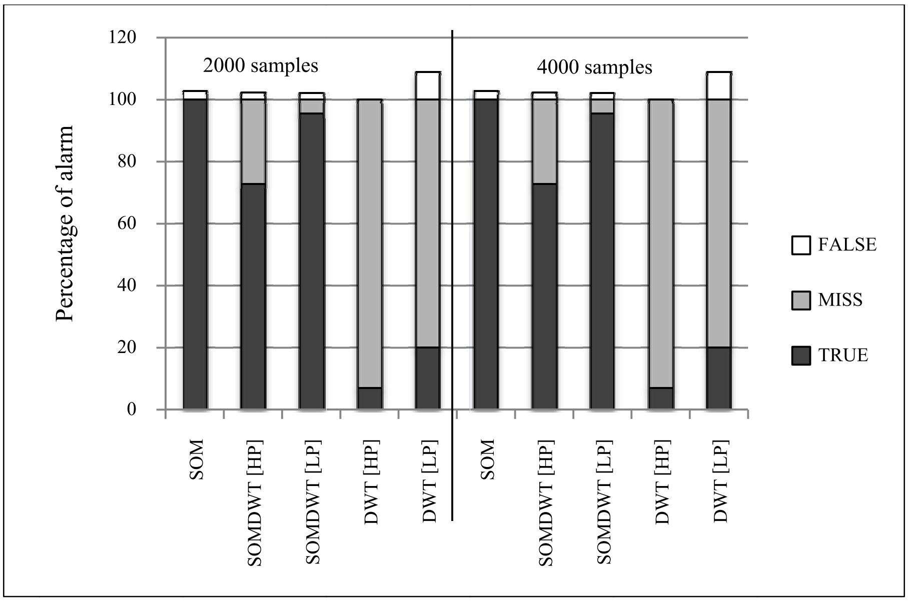


Figure 3.36 Detection rate in the bioorganic fertilizer dataset using the size of neurons of 10x10.

The results from the real-world dataset and bioorganic fertilizer dataset showed that our proposed algorithm, the integrated SOM and DWT algorithm performed as equally well as the SOM algorithm while using just half of the input data (using level 1 of DWT). This was because DWT was able to extract relevant data features without any significant loss in information thereby reducing wasted energy from transmitting all measurements to the base station. Hence, our results suggested that by applying DWT onto the sensor modes to achieve in-network data processing, the size of transmitted data can be reduced while still maintaining good anomaly detection abilities.

However, a variety of data characteristics can affect the anomaly detection performance of the integrated SOM and DWT algorithm as can be seen from the NAMOS dataset. Hence, a suitable setting of the algorithm, such as the size of training epochs, has to be considered carefully. In terms of the number of neurons, the more neurons used, the finer SOM's classification became, generally resulting in enhanced detection performance. However, the results in the real-world datasets and BOF dataset showed that there was no significant change in detection performance. In terms of the selection of wavelet coefficients, high pass coefficients were more suitable for detecting the changing points of the data, whereas low pass coefficients were more suitable for detecting the changing of trend of the data. These settings can be predetermined by considering the nature of the sensors deployed. For example, calibration errors in sensors can cause offset faults (whereby the measured value can differ from the true value by a constant), low battery voltage can cause a combination of noise and constant faults, while short faults can be caused by software error during communication and data logging (Sharma, Golubchik, and Govindan, 2010).

3.4 Summary

In this chapter, we evaluated the anomaly detection algorithms performance on different datasets, synthetic and real data with the synthetic faults (known faults) and the real-world datasets with the real faults (unknown faults). In terms of the true alarm rate, the proposed algorithm outperformed the SOM algorithm by up to 18% and DWT algorithm by up to 35% in presence of bursty faults. With sparse faults, the proposed algorithm can gain a true alarm rate up to 10% above the SOM algorithm alone and entirely outperform the DWT algorithm alone. Such gain in true alarm rates came with a marginal increase of false alarm rate.

In case of real-world datasets, we presented the anomaly detection in 4 different resources, NAMOS, INTEL, and 2 SensorScope (pdg2008 and SensorScope no.39 datasets) datasets. Our proposed algorithm with Haar as a mother wavelet can attain up to 99%, 100%, 83%, and 100% of true alarm rates in the NAMOS, INTEL, and 2 SensorScope datasets, respectively. Our proposed algorithm also performed as equally well as the SOM algorithm and outperformed the DWT algorithm by up to 15%, 100%, 17%, and 8% in the NAMOS, INTEL, and 2 SensorScope datasets, respectively.

In case of BOF dataset, our proposed algorithm with Haar as a mother wavelet using low pass coefficients can attain 95% of true alarm rates. Our proposed algorithm also performed as equally well as the SOM algorithm and outperformed the DWT algorithm by up to 75%.

The results showed that our proposed algorithm can maintain acceptable anomaly detection accuracy while using just half of the input data (using level 1 DWT).

CHAPTER IV

CONCLUSION AND FUTURE WORK

4.1 Conclusion

This thesis proposed an integration of a competitive learning method called the self-organizing map (SOM) and the discrete wavelet transform (DWT), to detect anomalies from series of data containing synthetic faults and faults obtained from real-world datasets. Our proposed algorithm, the integrated SOM and DWT algorithm, could help reduce wasted energy caused by transmitting all measurement data to the base station by applying the DWT algorithm onto the sensor modes in order to reduce size of transmitted data without losing the significant feature of the data. The original contributions and findings in this thesis can be summarized as follows.

4.1.1 Synthetic faults experiments

In the synthetic faults experiments, the results showed that the integration of SOM and DWT with Haar as a mother wavelet can attain 65% and 67% of true alarm rates in the case of bursty faults, and 69% and 80% of true alarm rates in case of sparse faults for synthetic and real data, respectively. In terms of the true alarm rate, the proposed algorithm outperformed the SOM algorithm by up to 18% and DWT algorithm by up to 35% in presence of bursty faults. With sparse faults, the proposed algorithm can gain a true alarm rate up to 10% above the SOM algorithm alone and entirely outperformed the DWT algorithm alone. Such gain in true alarm rates came with a marginal increase of false alarm rate.

4.1.2 Faults in real-world datasets

In the real-world datasets, the integration of SOM and DWT with Haar as a mother wavelet can attain up to 99%, 100%, 83%, and 100% of true alarm rates in the NAMOS, INTEL, SensorScope (pdg2008), and SensorScope (station no.39) datasets, respectively. Our proposed algorithm also performed as equally well as the SOM algorithm and outperformed the DWT algorithm by up to 15%, 100%, 17%, and 8% in the NAMOS, INTEL, SensorScope (pdg2008), and SensorScope (station no.39) datasets, respectively.

When reducing the number of training epochs, the proposed algorithm was directly affected. Hence, care must be taken when selecting the suitable number of training epochs. In the INTEL dataset and the SensorScope (station no.39) dataset, the proposed algorithm outperformed the DWT algorithm and performed equally well when compared to the SOM algorithm while using just half of the input data. In the SensorScope (pdg2008) dataset, the proposed algorithm outperformed the DWT algorithm but was slightly lower than the SOM algorithm.

By reducing the size of neurons, the proposed algorithm still obtained a true alarm rate up to 16%, 100%, 84%, and 17% higher than the DWT algorithm in NAMOS, INTEL, SensorScope (pdg2008), and SensorScope (station no.39) datasets, respectively. The proposed algorithm performed equally well as the SOM algorithm in the NAMOS, INTEL and SensorScope station no.39 datasets and only 2% lower than the SOM algorithm in the SensorScope (pdg2008) dataset. The reduction of the size of neurons did not show any significant change in detection performance.

4.1.3 Faults in the bioorganic fertilizer plant

In the BOF dataset, the proposed algorithm using low pass coefficients achieved a true alarm rate 5% less than the SOM algorithm while false alarm rate remained lower than 3%. The proposed algorithm using low pass coefficients can attain a true alarm rate of up to 75% more than the DWT algorithm alone. The effect of reducing the number of training epochs was not significant on the performance of SOM and the integrated SOM and DWT. In addition, the reduction of the size of neurons did not show any significant change in detection performance.

Our results suggested that the integration of SOM and DWT with Haar wavelet can lead to more effective anomaly detection. In particular, our results confirmed that the proposed algorithm can maintain acceptable anomaly detection accuracy while using just half of the input data (using level 1 DWT) instead of transmitting entire data. This could help reduce wasted energy caused by transmitting all measurement data to the base station.

However, since we did not have ground truth information about the faults incurred in these datasets, visual inspection and the histogram method were used to decide whether the data is normal or abnormal. Therefore, these methods are just heuristic estimation methods which may not coincide with the actual fault. Nevertheless, justifications of these faults can be made by consulting experts with domain knowledge on the information gathered from the environment under consideration (Sharma, Golubchik, and Govindan, 2010).

On the other hand, a variety of data characteristics can affect the anomaly detection in the integrated SOM and DWT algorithm as can be seen from the NAMOS dataset. Hence, a suitable setting of the algorithm, such as the size of training epochs, had to be considered carefully. In terms of the number of neurons, the more neurons used, the finer SOM’s classification became, generally resulting in enhanced detection performance.

4.2 FUTURE WORK

In the future, there are certain issues worthwhile investigating.

4.2.1 Increasing DWT level

The DWT obtains the hierarchical coefficients which can extract interesting the features of data. However, in our experiment we consider just the first level of the DWT coefficients. Considering other DWT coefficients level may be able to improve the anomaly detection algorithm performance.

4.2.2 Exploring other types of wavelets

To facilitate calculation by hand and allow comparison with the coefficients calculated from MATLAB program, we chose the Haar and Daubechies4 as mother wavelets. However, there are many types of the wavelets family which may affect the performance of the proposed anomaly detection algorithm.

4.2.3 Implementation on the sensor nodes

Another interesting direction is to investigate ways to identify and eliminate erroneous sensor readings directly at the sensor nodes (Liu and Zhou, 2010). which could help further reduce wasted energy from transmitting unwanted erroneous measurements to the base station.

4.2.4 Comparison with other data compression techniques

WSNs are resource constraint: limited power supply, bandwidth for communication, processing speed, and memory space. One possible way of achieve maximum utilization of those resource is applying data compression on sensor data (Kimura and Latifi, 2005); (Sadler and Martonosi, 2006). It could be better to find out the most suitable data compression algorithm for anomaly detection in WSNs.

4.2.5 Enhancing to fault predictability

The anomaly detection algorithm in this thesis can support detection when faults have already occurred. A worthwhile issue not only to be able to detect faults when they have already occurred but to predict them before a fault occurs. Such extension allows the user to take a suitable course of action to prevent the monitored environment before any significant damage occurs.

REFERENCES

Aquino, V.A., and Barria, J.A. (2001). Anomaly Detection in Communication Networks using Wavelets. **IEEE Proceedings in Communications**, December 2001, pp: 355-362.

Brychta, R. J., Tuntrakool, S., Appalsamy, M., and Robertson, D. (2007). Wavelet Methods for Spike Detection in Mouse Renal Synpathetic Nerve Activity, **IEEE Transactions on Biomedical Engineering**, January 2007, pp: 82-93.

Bruce, L.M., Cheriyadat, A., and Burns, M., (2003). Wavelets: Getting perspective. **Proceedings of Institute of Electrical and Electronics Engineers**, 2003, pp: 24-27.

Bruce, L. M., Koger, C. H., and Li, J. (2002). Dimensionality-Reduction of Hyperspectral Data using Discrete Wavelet Transform Feature extraction. **IEEE Transactions on Geoscience and Remote Sensing**, October 2002, pp: 2318-2338.

Brechet, L., Lucas, M-F., Doncarli, C., and Farina, D. (2007). Compression of Biomedical Signals with Mother Wavelet Optimization and Best-Basis Wavelet Packet Selection. **IEEE Transactions on Biomedical Engineering**, December 2007, pp: 2186-2192.

Barford, P., Kline, J., Plonka, D., and Ron, A. (2002). A signal analysis of network traffic anomalies. **Proceedings of the 2nd ACM SIGCOMM**, November 2002, pp.71–82.

Box, G. E. P., Jenkins, G. M., and Reinsen, G. C. (1994). Time Series Analysis: Forecasting and Control, 3rd Edition. **Prentice Hall**.

Bengio, Y., and Frasconi, P. (1996). An Input Output HMM Architecture. In **IEEE Transactions on Neuron Networks,** September 1996, pp: 1231–1249.

Barreto, G.A., Mota, J.C.M., Souza, L.G.M., Frota, R.A., and Aguayo, L. (2006). Condition Monitoring of 3G Cellular Network through Competitive Neural Models. **IEEE Transactions on Neural Networks**, September 2006, pp: 1064-1075.

Cordina, M., and Debono, C.J. (2008). Increasing Wireless Sensor Network Lifetime through the Application of SOM Neural Networks. **ISCCSP 2008**, March 2008, pp: 467-471.

Ciancio, A., Pattem, S., Otega, A., and Krishnamachari, B. (2006). Energy-Efficient Data Representation and Routing for Wireless sensor networks based on a Distributed Wavelet compression algorithm. **IPSN 2006**, April 2006, pp: 309-316.

Cheng, Y., Zhang, Y., Hu, J., and Li, L. (2007). Mining for Similarlities in Urban Traffic Flow using Wavelets. **Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference**, September 2007.

Doshi, R.A., King, R.L., and Lawrence, G.W. (2007). Wavelet-SOM in Feature Extraction of Hyperspectral Data for Classification of Nematode Species. **IEEE International Geoscience and Remote, Sensing Symposium (IGARSS**), July 2007, pp: 2818–2821.

Feather, F.E., Siewiorek, D., and Maxion, R. (1993). Fault Detection in an Ethernet using Anomaly Signature Matching. **ACM SIGCOMM’93**, 1993, pp: 279-288.

Guo, Y., Corke, P., Poulton, G., Wark, T., Bishop-Hurley, G., and Swain, D. (2006). Animal Behaviour Understanding using Wireless Sensor Networks. **31st IEEE Conference on Local Computer Networks**, November 2006, pp: 607-611.

Goh, H.G., Sim, M.L., and Ewe, H.T. (2007). Agriculture Monitoring. **Sensor Networks and Configuration**, 2007, Springer, pp: 439-462.

Hajji, H. (2003). Statistical Analysis of Network Traffic for Adaptive Faults Detection. **IEEE Transactions on Neural Network**, September 2003, pp: 1053-1063.

Ho, L.L., Cavuto, D.J., Papavassiliou, S., and Zawadzki, A.G. (2000). Adaptive and Automated Detection of Service Anomalies in Transaction-Oriented WAN’s: Network Analysis, Algorithms, Implementation, and Deployment. **IEEE Journal of Selected Areas in Communications**, May 2000, pp: 744-757.

Hood, C., and Ji, C. (1997). Proactive network fault detection. **IEEE Transactions Reliability**, 1997, pp: 333-341.

Hood, C.S., and Ji, C. (1997). Proactive network fault detection. **INFOCOM '9, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies**, April 1997 pp: 1147- 1555.

Huang, P., Feldmann, A., and Willinger, W. (2001). A Non-Intrusive Wavelet-Based Approach to Detecting Network Performance Problems. **Proceedings of the 1st ACMSIGCOMMWorkshopon Internet Measurment (IMW 2001)**, November 2001, pp: 213–227.

INTEL. (2004). **The Intel Lab Data. Data set available** at: http://berkeley.intel-research.net/labdata/

Kimura, N., and Latifi, S. (2005). A Survey on Data Compression in Wireless Sensor Networks, **International Conference on Information Technology: Coding and Computing (ITCC 2005)**, April 2005, pp: 8-13.

Kim, M.S., Kim, T., Shin, Y.J., Lam, S.S., and Powers, E.J. (2004). A Wavelet-Based Approach to Detect Shared Congestion. **ACM SIGCOMM Computer Communication Review**, 2004, pp: 293–306.

Kim, S.S., Reddy, A.L.N., and Vannucci, M. (2004). Detecting traffic anomalies through aggregate analysis of packet header data. **Proceedings of the 3rd International IFIP-TC6 Networking Conference**, May 2004, pp: 1047–1059.

Kailath, T., Ed. (1977). Linear Least-Squares Estimation. **Hutchison & Ross, Stroudsburg**, Pa.

Kaur, G., Saxena, V., and Gupta, J.P. (2010). Anomaly Detection in Network Traffic and Role of Wavelets. **IEEE Transactions on Instrumentation and Measurement**, April 2010, pp: 46-51.

Kwong, K.H., Wu, T.T., Sasloglou, K.,Stephen, B., Cao, D., Goh, H.G., Goo, S.K., Gilroy, M., and Tachtatzis, C. (2009). Implementation of herd management system with wireless sensor networks. **Joint International Agricultural Conference (JIAC2009)**, July 2009.

Li, X., Deng, Y., and Ding, L. (2008). Study on Precision Agriculture Monitoring Framework Based on WSN. **The 2nd International Conference on Anti-counterfeiting, Security and Identification**, August 2008, pp: 182-185.

Laiho, J., Raivio, K., Lehtimaki, P., Hatonen, K., and Simula O. (2005). Advanced Analysis Methods for 3G Cellular Networks. **IEEE Transactions on Neural Networks**, 2005, pp: 930-942.

Laiho, J., Kylvaja, M., and Hoglund, A. (2002). Utilization of Advanced Analysis Methods in UMTS Networks. **IEEE Vehicular Technology Conference**, May 2002, pp: 726-730.

Lee, M.H., and Choi, Y.H. (2008). Fault detection of wireless sensor networks. **Computer Communications**, 2008, pp: 3469-3475.

Lichodzijewski, P., Zincir-Heywood, A., and Heywood, M. (2002). Dynamic Intrusion Detection using Self-Organizing Maps. **Proceedings of the 14th CITSS**, May 2002.

Lakhina, A., Crovella, M., and Diot, C. (2004). Diagnosing network wide traffic anomalies. **Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’04)**, September 2004, pp: 219–230.

Li, X-L., Zhang, J-W., and Fang, W-H. (2009). The Research of Data Compression Algorithm Based on Lifting Wavelet Transform for Wireless Sensor Network. **International Conference on Apperceiving Computing and Intelligence Analysis**, October 2009, pp: 228-233.

Liu, J.F., and Zhou, N. (2010). Localization anomaly detection for wireless sensor networks, **IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)**, October 2010, pp: 644-648.

Maxion, R., and Feather, F.E. (1990). A Case Study of Ethernet Anomalies in a Distributed Computing Environment. **IEEE Transactions Reliability**, October 1990, pp: 433-443.

NAMOS. 2006. **Networked Aquatic Microbial Observing System. Data set available** at: http://robotics.usc.edu/∼namos/data/jr aug 06/.

Paladina, L., Paone, M., Jellamo, G., and Puliafito, A. (2007). Self-Organizing Maps for Distributed Localization in Wireless Sensor Networks. **12th IEEE Symposium on Computers and Communications**, July 2007, pp:1113-1118.

Postalcıoglu, S., Erkan, K., and Bolat, E.D. (2007). Implementation of Intelligent Active Fault Tolerant Control System. **Springer-Verlag Berlin Heidelberg 2007**, pp. 804–812.

Rajasegarar, S., Leckie, C., and Palaniswami, M. (2008). Anomaly Detection in Wireless Sensor Networks. **IEEE Wireless Communications**, 2008, pp: 34-40.

Ramanathan, N., Balzano, L., Burt, M., Estrin, D., Kohler, E., Harmon, T., Harvey, C., Jay, J., Rothenberg, S., and Srivastava, M. (2006). **Rapid Deployment with Confidence: Calibration and Fault Detection in Environmental Sensor Networks.** Tech. Rep. 62, Cens. April.

Ramadas, M., Ostermann, S., and Tjaden, B. (2003). Detecting Anomalous Network Traffic with Self-Organizing Maps. **Proceedings of the RAID**, September 2003, pp: 36-54.

Sukkhawatchani, P., and Usaha, W. (2008). Performance Evaluation of Anomaly Detection in Cellular Core Networks using Self-Organizing Map. **5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology**, May 2008, pp: 361-364.

Sharma, A.B., Golubchik, L., and Govindan, R. (2010). Sensor Faults: Detection Methods and Prevalence in Real-World Datasets. **Transactions on Sensor Networks**, 2010, pp: 1-34.

SENSORSCOPE. 2006. **The SensorScope Lausanne Urban Canopy Experiment (LUCE) Project. Data set available** at: http://sensorscope.epfl.ch/index. php/ LUCE.

Sadler, C.M., and Martonosi, M. (2006). Data Compression Algorithm for energy-constrained Devices in Delay Tolerant Networks, **Proceedings of the 4th International Conference on Embedded Networked Sensor Systems**, November 2006, pp: 265-278.

Thottan, M., and Chuanyi, J. (2003). Anomaly Detection in IP Network. **IEEE Transactions on Signal Processing**, 2003, pp: 2191-2204.

Walker, J.S. (1999). A Preimer on Wavelets and their Scientific Applcations. **Chapman & Hall**/CRC,1999.

Xu, Z., and Zhao, Q. (2002). A Novel Approach to Fault Detection and Isolation Based on Wavelet Analysis and Neural Network. **Electrical and Computer Engineering**, May. 2002, pp: 572–577.

Yadaiah, N., and Ravi, N. (2007). Fault Detection Techniques for Power Transformers. **Industrial & Commercial Power Systems Technical Conference**, 2007, pp: 1- 9.

Yang, X., and Tommy, W.S.C. (2010). Efficient Self-Organizing Map Learning Scheme using Data Reduction Preprocessing, **Proceedings of the World Congress on Engineering 2010**, July 2010.

Zheng, J., and Hu, M. (2005). Detection of TCP Attacks Using SOM with Fast Nearest-Neighbor Search. **WSEAS International Conference on Neural Networks**, 2005, pp: 176-182.

APPENDIX A

List of Publications

List of Publications

Siripanadorn, S., Hattagam, W., and Teaumroong, N. (2010). **Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets.** The 9th International Conference on Applied Computer Science (WSEAS), Japan, October 2010.

Siripanadorn, S., Hattagam, W., and Teaumroong, N. (2010). **Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets.** The International Journal of Communications, December 2010.

Anomaly Detection using Self-Organizing Map and Wavelets in Wireless Sensor Networks

SUPAKIT SIRIPANADORN[1], WIPAWEE HATTAGAM[2], NEUNG TEAUMROONG[3]
[1,2]School of Telecommunication Engineering, Institute of Engineering
[3]School of Biotechnology, Institute of Agriculture
Suranaree University of Technology
111 University Avenue, Muang, Nakhon Ratchasima 30000
THAILAND
[1]architect_ton@hotmail.com [2]wusaha@ieee.org [3]neung@sut.ac.th

Abstract: - Wireless Sensor Networks (WSNs) have been developed and extensively applied in agriculture monitoring to monitor and collect various physical attributes within a specific area or environment of interest. Data readings from the sensors may be abnormal due to the sensors themselves such as limited battery power, onboard processing capability, sensor malfunction, or noise from the communication channel. It is thus, important to detect such data anomalies available in WSNs to determine a suitable course of action. The underlying aim of this paper is therefore to propose an anomaly detection scheme which is able to detect anomalies accurately by means of exploiting both time and frequency characteristics of the data signals. The contribution of this paper centers on anomaly detection by using Discrete Wavelet Transform (DWT) combined with a competitive learning neural network called self-organizing map (SOM) in order to accurately detect abnormal data readings are collected from WSNs.

Key-Words: - anomaly detection, wavelets, discrete wavelet transform, self-organizing map, wireless sensor networks, agriculture monitoring

1 Introduction

Wireless sensor networks (WSNs) have been recently deployed in many areas of agriculture to increase yield and prevent outbreaks such as in hydroponics and paddy fields, fertilizer composting process, and livestock monitoring. However, these applications rely mainly on manually measuring and controlling the parameters such as moisture, homogeneity, temperature, pH, oxygen, soil nutrients, etc., which is both time consuming and laborious. Autonomous monitoring devices such as WSNs therefore warrant potential use in agriculture monitoring.

A WSN is a wireless network that consists of distributed autonomous devices using sensors to cooperatively monitor or collect environmental conditions at different locations. Several measurements can be collected from the WSN. The collected measurements from the WSN may be affected by anomalies in the sensor network such as faulty sensors, faulty communication between sensors or actual abnormal physical measurements. With the huge amount of data continually collected from the WSN, it becomes increasingly difficult to detect anomalies in the data measurements. Therefore, anomaly detection techniques are necessary to automatically detect faults and alert the system controller to take suitable action.

Research emphasizing on anomaly detection in communication networks has progressed in recent years, e.g. in IP networks [1], in cellular mobile networks [2]. There are also works on fault and anomaly detection in wireless sensor networks (WSNs) [3]. Ref [4] presented a dynamic model of wireless sensor networks (WSNs) based on recurrent neural networks (RNNs) and used it for fault detection at the sensor node.

Another mechanism commonly used for anomaly detection is a competitive learning method called self-organizing map (SOM) [5], [6]. SOM has several beneficial features which make it a useful tool in data mining. It follows the probability density function of the data and is, thus, an efficient clustering and quantization algorithm. The most important feature of the SOM is the visualization property.

However, SOM has some weaknesses where it extracts relevant features of the data only in the time domain. In many scenarios, features of the data extracted from both time and frequency domain can be used to further enhance anomaly detection [7]. This can be achieved by the Discrete Wavelet Transform (DWT). Wavelets have been extensively employed for anomaly and fault detection in many applications [8]. DWT has also been integrated with SOM to detect faults [9], [10]. In particular, feature vectors of the faults have been constructed using DWT, sliding window and a statistical analysis. Classification of the feature vectors was obtained by using SOM.

To the best of our knowledge, DWT and SOM have not yet been applied to anomaly detection in WSNs. Therefore, the underlying aim of this paper is to propose an anomaly detection algorithm which determines the wavelet transform, and detects the abnormality of the sensor readings by training the SOM using the wavelet coefficients.

2 Anomaly Detection

The first step involves selecting the parameters to be monitored and grouping them together in a pattern vector $\mathbf{x}^{\mu} \in \Re$, $\mu = 1,\ldots,N$,

$$x^{\mu} = \begin{pmatrix} x_1^{\mu} \\ x_2^{\mu} \\ \vdots \\ x_n^{\mu} \end{pmatrix} = \begin{pmatrix} \mathrm{KPI}_1^{\mu} \\ \mathrm{KPI}_2^{\mu} \\ \vdots \\ \mathrm{KPI}_n^{\mu} \end{pmatrix}, \qquad (1)$$

where μ is the observation index, n is the number of KPIs chosen to monitor the environmental condition.

2.1 Self-Organizing Map

Competitive neural models such as the self-organizing map (SOM) [1], [11] are able to extract statistical regularities from the input data vectors and encode them in the weights without supervision. It maps a high-dimensional data manifold onto a low-dimensional, usually two-dimensional, grid or display.

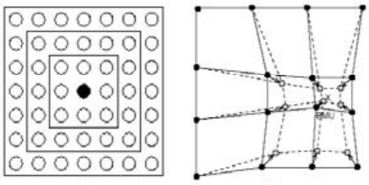
The basic SOM consists of a regular grid of map units or neurons as shown in Fig 1(a). Each neuron, denoted by i (depicted by the black dot), has a set of layered neighboring neurons (depicted by the white dots) as shown in Fig 1(a).

Neuron i maintains a weight vector $\mathbf{m}_i$. In order to follow the properties of the input data, such vector is updated during the training process. For example, Fig.1(b) shows a SOM represented by a 2-dimensional grid of 4×4 neurons. The dimension of each vector is equal to the dimension of the input data. In the figure, a vector of input data (marked by x) is used to train the SOM weight vectors (the black dots). The winning neuron (marked by BMU) as well as its 1-neighborhood neurons, adjust their corresponding vectors to the new values (marked by the gray dots).

The SOM is trained iteratively. In each training step, one sample vector $\mathbf{x}$ from the input data set is chosen. The distances between the sample data and all of weight vectors in the SOM are calculated using some distance measure. Suppose that at iteration t, neuron i whose weight vector $\mathbf{m}_i(t)$ is the closest to the input vector $\mathbf{x}(t)$. We denote such weight vector by $\mathbf{m}_c(t)$ and refer to it as the Best-Matching Unit (BMU), that is

$$\|x(t) - m_c(t)\| = \arg \min_{\forall i} \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \qquad (2)$$

where $\|\cdot\|$ is the Euclidian distance.



(a) (b)

Fig. 1 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3- neighborhoods, (b) SOM updates the BMU with 1-neighborhood.

Suppose neuron i is to be updated, the SOM updating rule for the weight vector of neuron i is given by

$$m_i(t+1) = m_i(t) + \alpha(t) h_c(i,t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \qquad (3)$$

where t is the iteration index, $\mathbf{x}(t)$ is an input vector, $\alpha(t)$ is the learning rate, $h_c(i,t)$ is the neighborhood function of the algorithm. The Gaussian neighborhood function may be used, that is

$$h_c(i,t) = \exp\left(- \frac{\|r_c(t) - r_i(t)\|^2}{2\sigma^2(t)} \right) \qquad (4)$$

where $r_i(t)$ and $r_c(t)$ are the positions of neurons i and the BMU c respectively, and $\sigma(t)$ is the radius of the neighborhood function at time t. Note that $h_c(i,t)$ defines the width of the neighborhood. It is necessary that $\lim_{t\to\infty} h_c(i,t) = 0$ and $\lim_{t\to\infty} \alpha(t) = 0$ for the algorithm to converge [1], [11].

2.1.1 Anomaly Detection

2.2 Discrete Wavelet Transform

DWT is a mathematical transform that separates the signal into fine scale information known as detail coefficients, and rough-scale information known as approximate coefficients. Its major advantage is the multi-resolution representation and time-frequency localization property for signals. Usually, the sketch of the original time series can be recovered using only the low-pass-cut off decomposition coefficients; the details can be modeled from the middle-level decomposition coefficients; the rest is usually regarded as noises or irregularities. The following equations describe the DWT decomposition process:

$$a_{j+1}^{DWT}(k)=\sum_n h_0(n-2k)a_j^{DWT}(k) \qquad (5)$$

$$d_{j+1}^{DWT}(k)=\sum_n g_0(n-2k)a_j^{DWT}(k)\ , \qquad (6)$$

where the broad scale, or *approximation*, coefficients a_{j+1}^{DWT} are convolved separately with h_0 and g_0, the wavelet function and scaling function, respectively; n is the time scaling index, k is the frequency translation index for wavelet level j. The resulting coefficient is down-sampled by 2. This process splits a_{j+1}^{DWT} roughly in half, partitioning it into a set of fine scale, or *detail* coefficients d_{j+1}^{DWT} and a coarser set of approximation coefficients a_{j+1}^{DWT} [12].

Therefore, DWT is powerful in encoding the finer resolution of the original time series with its hierarchical coefficients. Furthermore, DWT can be computed efficiently in linear time, which is important while dealing with large datasets.

2.3 Integration of SOM and DWT

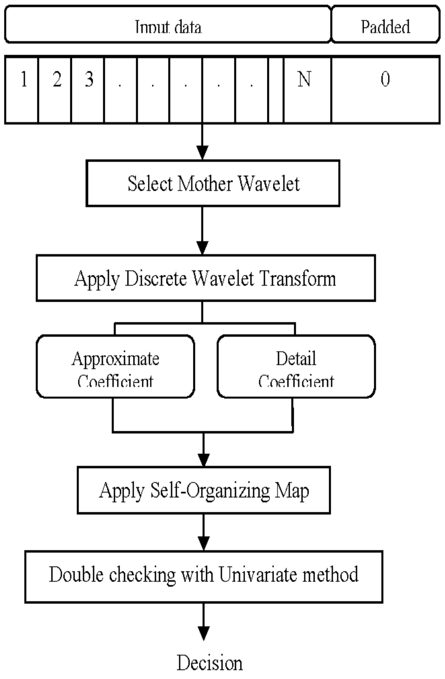


Fig.2 The integration of the SOM and DWT algorithm diagram.

In the integration of SOM and DWT algorithm, the DWT algorithm is used as an input data preprocessor of the SOM algorithm in order to reduce size of data without losing any significant feature of the data. The input data will be padded with zero if its length is odd data. After obtaining the wavelet coefficients, these coefficients will be fed to the SOM algorithm which can be divided into 2 sets. Each set contains both approximate and detail coefficients. The first set which is obtained from noiseless data, will be used to train the SOM algorithm and the second set will be used to test the SOM algorithm, respectively. Then the detected results will be double checked by using the univariate method [5].

2.4 Anomaly Detection

A new observation data set can be considered abnormal if the distance between the weight vector of the winning neuron and the new state vector, given by

$$e^{\mu}=\left\|\mathbf{x}^{new}-\mathbf{m}_c^{\mu}\right\| \qquad (7)$$

is greater than a certain percentage $p=1-\alpha$ of the distances in the distance distribution profile. That is,

IF $e^{\mu}\in\left[e_p^-,e_p^+\right]$,

THEN $\mathbf{x}^{new}$ is NORMAL

ELSE $\mathbf{x}^{new}$ is ABNORMAL. (8)

Equation (8) is referred to as the global decision. In [13], an addition of local decisions of each KPIs is presented. Suppose that a data vector $\mathbf{x}^{new}$ is considered abnormal by the global decision. Then in the local anomaly detection, the absolute value of error in each component of the error vector is then computed by

$$\left|\mathbf{E}^{new}\right|=\begin{pmatrix}\left|x_1^{\mu}-m_{c,1}^{\mu}\right|\\ \left|x_2^{\mu}-m_{c,2}^{\mu}\right|\\ \vdots\\ \left|x_n^{\mu}-m_{c,n}^{\mu}\right|\end{pmatrix}. \qquad (9)$$

The error in each KPI is then compared to the interval of normality component-by-component, and the anomaly decision is carried out as in (8).

3 Experiment Results

In this section, we evaluated the performance of the proposed integration of SOM and DWT algorithm by detecting anomalies in series of synthetic data and actual data collected from a wireless sensor network.

In the experiment, we generated the synthetic input data from a normal distribution N(0,1) and synthetic faults by additive white Gaussian noise (AWGN) with power 25 dBW generated from MATLAB. We used such fault because its statistical similarity to the synthetic input data thus, it is more difficult to be detected. Therefore, we can evaluate the performance of the algorithms under ambiguous faults. The amount of faults can be represented by the notation n/s, where "n" is the amount of faults per series and "s" is the amount of series of faults, resulting in the total amount of n×s faults. The generated faults added to the input data ranged from bursty (20/10) to sparse (1/10). The exact positions of the faults incurred in the input data were predetermined and was later used to detect true and false alarms. In the experiment using real data, we have chosen 2 parameters, namely temperature and moisture, as KPIs collected from samples of compost in a bioorganic fertilizer plant. In this paper, the data of the 2 KPIs at the WSNs were collected every 5 minutes for 3 days.

We compared 3 anomaly detection methods:

1. SOM algorithm
2. DWT algorithm
3. Integration of SOM and DWT algorithm

We measured 2 performance metrics: 1) the *true alarm rate* which is defined by the number of detected true anomalies over the total number of true anomalies in the data set; and 2) the *false alarm rate* which is defined by the number of detected false alarms over the total number of detected anomalies.

In the DWT algorithm, we used the threshold in (11) in order to decide whether the data is normal or abnormal.

$$\sigma_w = median(\left|d_1 - \overline{d_1}\right|) \qquad (10)$$

$$T_w = \sigma_w \sqrt{2\log_e(N)}, \qquad (11)$$

where N is the size of data and $\overline{d}_1$ is the sample mean of the level 1 detail coefficients [12].

This threshold was calculated from the low pass and high pass coefficients from the assumed normal data by using Haar and Daubechies4 mother wavelets. The Haar and Daubechies4 wavelets were used because they are relatively easy to cross-check by hand with computed coefficients from MATLAB program. Hence, we can compare the position of each coefficient with the actual fault position. After the threshold calculation, the set of coefficients which are obtained from the DWT of the noisy data will be compared with the threshold, coefficient by coefficient. For the real data scenario, the data was normalized by equation (12) before being processed by the DWT to eliminate potential outliers.

$$\text{Norm(Data)} = \frac{\text{(Data)} - \text{mean(Data)}}{\text{variance(Data)}} \qquad (12)$$

If the absolute value of the coefficient is greater than the computed threshold, an anomaly is said to be detected.

In the SOM algorithm and the proposed **integration of SOM and DWT algorithm**, the initial value for learning rate in the SOM part was set to η_0 = 0.9, and gradually reduced to $\eta_T = 10^{-5}$, in order to guarantee convergence [14]. The number of training epochs was set to 50 because longer training epochs tend to over train the SOM [6]. The confidence interval was set to 99% (K=2.57). We used a Gaussian neighborhood function because the distribution of the collected data after the normalization fits well to the Gaussian distribution. The 30×30 size of neurons was used. Figures 3 and 4 show that the anomaly detection in SOM algorithm and the integrated SOM and DWT algorithms improve as the number of neurons is increased. This suggests that the more neurons used, the "finer" SOM's classification becomes resulting in enhanced detection performance. However, at neuron size 50×50, the SOM requires much longer training time with a marginal improvement in the detection performance. Therefore, The 30×30 size of neurons was selected to train and test the SOM. We also improved the SOM algorithm by double checking with the univariate method in order to reduce the false alarm rate [5]. To obtain accurate results, each algorithm was repeated for 70 runs which gave the best accuracy as shown in Table1.

Table 1. Accuracy results of the true alarm rate obtained by feeding synthetic input data to the 30×30 neuron SOM algorithm.

Runs	True Alarm rate	Deviation[1]
1	62.00	-
10	59.50	0.040
20	57.65	0.031
30	58.17	0.009
40	57.68	0.008
50	57.82	0.002
60	57.88	0.004
70	58.14	0.001
80	58.06	0.001
90	58.17	0.001
100	58.27	0.001

[1] $\text{Deviation} = \left|1 - \text{Ratio}\right|$

$$\text{Ratio} = \frac{\text{True alarm of the current iteration}}{\text{True alarm of the previous iteration}}$$

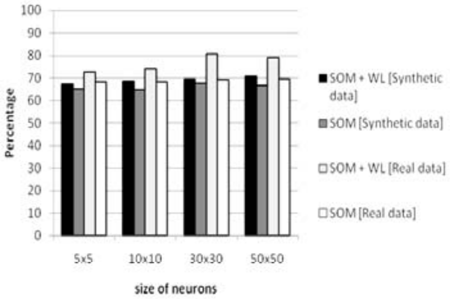


Fig. 3, True alarm rates with different size of neurons in the sparse faults case.

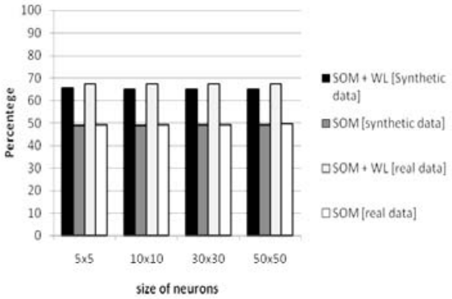


Fig. 4, True alarm rates with different size of neurons in the bursty faults case.

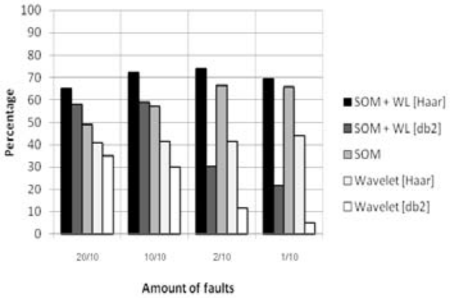


Fig.5, True alarm rates with synthetic data.

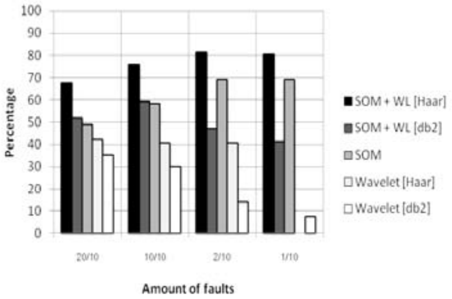


Fig.6, True alarm rates with real data.

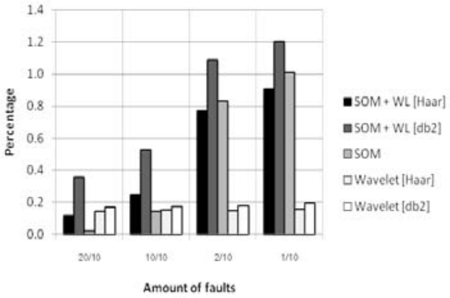


Fig.7, False alarm rates with synthetic data.

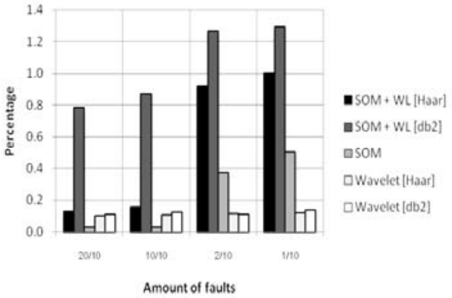


Fig.8, False alarm rates with real data.

In the proposed integration of SOM and DWT algorithm, we improved the performance of the SOM part of the algorithm by replacing the input synthetic data with low pass and high pass coefficients obtained from the DWT which used Haar and Daubechies4 as mother wavelets. In the case of actual data, we normalized all of data by (12) before passing it through the DWT process as well. The coefficients obtained from the noiseless data were used to train the SOM. To test the SOM, the synthetic faults previously described were added to the original set of noiseless data.

To evaluate the performance of all algorithms, the results of each algorithm were compared to the (known) fault positions which were added into the input data. In particular, when an anomaly was detected then its position was compared with the (known) fault position. If this position existed, then the anomaly detected was a true alarm; otherwise, it was a miss. On the other hand, if an anomaly was detected but the (known) fault position did not exist, then the anomaly was a false alarm.

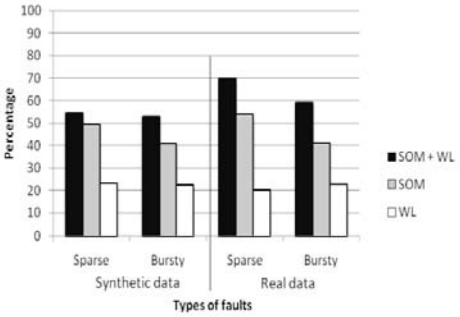


Fig. 9, True alarm rate with 10 dBW AWGN faults.

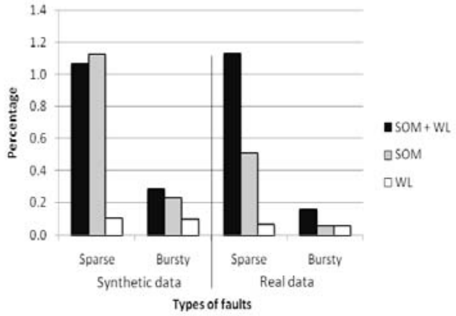


Fig. 10, False alarm rate with 10 dBW AWGN faults.

Fig. 5 and Fig. 6 show the percentage of true alarm rate averaged over 70 runs, as the function of the amount of faults added into the input data. Note that the proposed integrated SOM and DWT algorithm which used Haar as a mother wavelet gives the best performance over other algorithms. This is because the DWT with Haar wavelet can detect changing points. In particular, the Haar wavelet uses 2 adjacent input data to compute a coefficient whereas the Daubechies4 uses 4 adjacent input data to compute a coefficient. However, Daubechies4 gave a lower performance than Haar because each coefficient was computed from an average over 4 input data. If a fault occurred in 1 of these 4 data, such fault will be averaged with the remaining 3 normal data resulting in a coefficient with an absolute value possibly lower than the decision threshold. Consequently, the true alarm rate is reduced. On the other hand, the Haar wavelet only uses 2 adjacent data to compute 1 coefficient. Thus, the true alarm rate is significantly higher than that of Daubechies4. The integrated SOM and DWT algorithm using Haar also outperforms the SOM algorithm. This is because in the Haar case, the coefficients obtained were transformed from two adjacent data. Therefore, if some data was faulty or differed greatly from the data nearby, this coefficient can detect such anomaly. On the other hand, the SOM algorithm directly checked the data one by one to detect an anomaly. If the data was faulty but had a small magnitude, then this fault may not be detected, and consequently the true alarm rate was reduced. Note that the DWT algorithm has the lowest performance because the decision threshold in (11) is rather conservative. Furthermore, the threshold is fixed throughout the detection and the algorithm does not have any double checking method.

Fig.7 and Fig. 8 show the false alarm rate results in the synthetic and real data experiments, respectively. Note that most results have low false alarm rates, i.e., less than 1 % except in the case of sparse faults due to the increased detection difficulty.

The integration of SOM and Daubechies4 DWT also gave a weak performance due to the reasons previously explained. All these results show that the integration of SOM and DWT with Haar as a mother wavelet outperforms the SOM algorithm and DWT method as it can achieve upto 65% and 67% of true alarm rates in case of bursty faults for synthetic and real data, respectively. As for sparse faults, the proposed algorithm can achieve upto 69% and 80% true alarm rates for synthetic and real data, respectively. In addition, the false alarm rate is 0.11% and 0.13% in case of bursty faults and 0.91% and 1% in case of sparse faults with synthetic and real data, respectively.

DWT to reduce transmitted data: The proposed integration of SOM and DWT algorithm with Haar wavelet outperformed the SOM algorithm and the DWT algorithm alone. Our results suggest that the proposed integrated SOM and DWT anomaly detection scheme can be deployed in a resource-constrained network such as a WSN. In particular, the DWT using Haar wavelet can be implemented at the sensor nodes as a data preprocessor to reduce the amount of data to be transmitted by at least half (for one-level DWT). Since energy consumption is critical in WSNs, such distributed in-network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN [3] while still maintaining acceptable anomaly detection accuracy.

Fig. 9 and Fig. 10 show the effect of the decreasing of AWGN noise power from 25dBW to 10 dBW in both synthetic and real data scenarios. Though the anomaly detection is more difficult, the proposed integrated SOM and DWT still consistently outperforms the other two methods in terms of true alarm rate but with marginal increase in the false alarm rate as tradeoff.

4 Conclusion

This paper proposed an integration of a competitive learning method called the self-organizing map (SOM) and the discrete wavelet transform (DWT), to detect anomalies in synthetic and real data collected from WSN deployed in a bioorganic fertilizer plant.

The results show that the integration of SOM and DWT with Haar as a mother wavelet can attain 65% and 67% of true alarm rates in the case of bursty faults, and 69% and 80% of true alarm rates in case of sparse faults for synthetic and real data, respectively. The proposed algorithm outperforms the SOM algorithm and DWT algorithm. In addition, the false alarm rate is 0.11% and 0.13% in case of bursty faults and 0.91% and 1% in case of sparse faults for synthetic and real data, respectively. The proposed algorithm can maintain acceptable anomaly detection accuracy while using just half of the input data. Our results suggest that the integration of SOM and DWT with Haar wavelet can lead to a more effective anomaly detection which reduces human operator's troubleshooting efforts.

In the future, we plan to extend our work to investigate anomaly detection with actual faults obtained from the bioorganic fertilizer plant environment, and study its performance by increasing the DWT level and considering other different types of wavelets.

5 Acknowledgment

The authors would like to thank Assoc. Prof. Dr. Boon Hee Soong and Mr. Edwin Chan of Intelligent Systems Laboratory, Nanyang Technological University, Singapore for their assistance and suggestions on the sensor development and implementation.

References:

[1] M. Thottan and J. Chuanyi, Anomaly detection in IP network, *IEEE Trans. on signal processing*, Vol.51, No.8, 2003.

[2] J. Laiho, K. Raivio, P. Lehtimaki, K. Hatonen, and O. Simula, Advanced Analysis Methods for 3G Cellular Networks, *IEEE Trans. Neural Networks*, Vol.4, No.3, 2005, pp. 930-942.

[3] S. Rajasegarar, et al., Anomaly Detection in Wireless Sensor Networks, *IEEE Wireless Communications*, Vol.15, No.4, 2008, pp. 34-40.

[4] A.I. Moustapha, and R.R. Selmic, Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection, *IEEE Trans. on Instrumentation and Measurement*, Vol. 57, No. 5, 2008, pp. 981-988.

[5] G.A. Barreto, J.C.M. Mota, L.G.M. Souza, R.A. Frota, and L. Aguayo, "Condition monitoring of 3G cellular network through," *IEEE Trans. Neural Networks*, Vol. 16, No. 5, 2006, pp. 1064-1075.

[6] P. Sukkhawatchani and W. Usaha., Performance Evaluation of Anomaly Detection in Cellular Core Networks using Self-Organizing Map, *Proceeding of ECTI-CON 2008*, Vol.1, 2008, pp. 361-364.

[7] V.A. Aquino and J.A. Barria, Anomaly detection in communication Networks using wavelets, *IEEE Proc.-Commun*, Vol.148, No.6, 2001, pp. 355-362.

[8] N. Yadaiah, and Nagireddy Ravi, Fault detection techniques for power transformers, *Industrial & Commercial Power Systems Technical Conference*, 2007, pp. 1- 9.

[9] Z. Xu and Q. Zhao, A novel approach to fault detection and isolation based on wavelet analysis and neural network, *Electrical and Computer Engineering*, Vol. 1, 2002, pp. 572–577.

[10] S. Postalcıoglu, K. Erkan, and E.D. Bolat, Implementation of Intelligent Active Fault Tolerant Control System, *Springer-Verlag Berlin Heidelberg 2007*, pp. 804–812.

[11] X.Li, Y.Deng and L. Ding, Study on precision agriculture monitoring framework based on WSN, *Anti-counterfeiting, Security and Identification,* 2008, pp. 182–185.

[12] R. J. Brychta, et al, Wavelet Methods for Spike Detection in Mouse Renal Synpathetic Nerve Activity, *IEEE Transactions on Biomedical Engineering*, Vol.54, No.1, 2007, pp. 82-93.

[13] H. Hajji, et al., Statistical analysis of network traffic for adaptive faults detection, *IEEE Trans. on Neural Network*, Vol. 16, No. 5, 2003, pp. 1053-1063.

[14] G. A. Barreto, Joao C. M. Mota and Luiz G. M. Souza., Condition Monitoring of 3G Cellular Networks Through Competitive Neural Models, *IEEE Traansactions on Neuron Networks,* Vol.16, No.5, 2005, pp. 1064-1075.

Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets

S. Siripanadorn, W. Hattagam, N. Teaumroong

Abstract**—**This paper proposes an anomaly detection scheme which is able to detect anomalies accurately by employing only important features of data signals, instead of using all the sensor data traces. The contribution of this paper centers on anomaly detection by using Discrete Wavelet Transform (DWT) combined with a competitive learning neural network called self-organizing map (SOM) in order to accurately detect abnormal data readings while using just half of the data size. Experiment results from synthetic and real data injected with synthetic faults collected from a WSN show that the proposed algorithm outperforms the SOM algorithm by up to 18% and DWT algorithm by up to 35% in presence of bursty faults with marginal increase of false alarm rate. Furthermore, in the real-world datasets experiments show that our proposed algorithm can maintain acceptable anomaly detection accuracy as well as the SOM algorithm while using just half of the input data.

Keywords**—**Anomaly Detection, Discrete Wavelet Transform, Self-Organizing Map, Wireless Sensor Networks, Agriculture Monitoring.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been recently deployed in many areas of agriculture to increase yield and prevent outbreaks such as in hydroponics and paddy fields, fertilizer composting process, and livestock monitoring. However, these applications rely mainly on manually measuring and controlling the parameters such as moisture, temperature, pH, oxygen, soil nutrients, etc., which are both time consuming and laborious. Autonomous monitoring devices such as WSNs therefore warrant potential use in agriculture monitoring.

A WSN is a wireless network that consists of distributed autonomous devices using sensors to cooperatively monitor or collect environmental conditions at different locations. Several measurements can be collected from the WSN. The collected measurements from the WSN may be affected by anomalies in the sensor network. With the huge amount of data continually collected from the WSN, it becomes increasingly difficult to detect anomalies in the data measurements. Therefore, anomaly detection techniques are necessary to automatically detect faults and alert the system controller to take suitable action.

Research emphasizing on anomaly detection in communication networks has progressed in recent years,e.g., in network traffic [1], [2], [3], in IP networks [4], in cellular mobile networks [5]. In general anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [1]. Abnormal data patterns can be caused by faulty sensors in the network or unusual phenomena in the monitored domain.

Anomalies caused by faulty sensor communications are presented in [6]. They proposed a distributed algorithm for detecting and isolating faulty sensor nodes in WSNs. Each sensor node identifies its own status based on local comparisons of sensed data with thresholds. Ref. [7] applied 4 different anomaly detection techniques for different types of faults obtained in the real–world datasets, namely, NAMOS [8], INTEL [9] and SensorScope [10]. They classified these faults into 3 types, i.e., noise faults, short faults and constant faults. This research suggested that there is presently no known anomaly detection method suitable for every type of faults.

Another application of anomaly detection is an unusual phenomenon in the monitored domain. Erroneous measurements may occur as a result of transducers drifting out of calibration, or from faults introduced by harsh environmental conditions. In a large network it is extremely difficult and time consuming to detect these erroneous measurements manually. In addition, energy is wasted in the network when forwarding the unwanted erroneous measurements to the base station for analysis. One solution to alleviate network energy consumption is to reduce the amount of data that needs to be communicated through the network. Energy is critical in WSNs, therefore anomaly detection methods in WSN must not only perform well but also demand low energy consumption. Distributed in-network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN [11]. Our work is motivated by this concept. In particular, we focus on reducing the amount of transmitted data by in-network processing for anomaly detection at the base station.

This paper considers anomalies caused by unusual phenomenon and faulty sensors. To detect these anomalies, a dynamic data classification scheme such as data mining method could be useful.

Data mining is an expanding area of research in artificial neural network and information management whose objective is to extract relevant information from large databases. One particular method, called the self-organizing map (SOM), has several beneficial features which make it a useful tool in data mining. In particular, it follows the probability density function of the data and is, thus, an efficient clustering and quantization algorithm. The most important feature of the SOM in data mining is the visualization property [12].

SOM has been applied for anomaly detection in communication networks [13], [14], [15] as well as WSNs [16]. Ref. [16] focuses on evaluating the position of sensors in a WSN, or the localization problem. Their localization technique is based on a simple SOM, implemented on each sensor node. The main advantages of their solution are the limited storage and computing costs. However, SOM requires processing time which increases with the size of input data. To reduce the input data size, features of the data can be extracted without losing the significant data can be used for anomaly detection. This can be achieved by the Discrete Wavelet Transform (DWT). Wavelets have been extensively employed for anomaly [17] and fault detection [18]. DWT has also been integrated with SOM to detect system faults [19], [20]. In particular, feature vectors of the faults have been constructed using DWT, sliding windows and a statistical analysis. Classification of the feature vectors was obtained by using SOM.

To the best of our knowledge, DWT and SOM have not yet been applied for anomaly detection in WSNs. Therefore, the underlying aim of this paper is to propose an anomaly detection algorithm which determines the discrete wavelet transform, and detects the abnormality of the sensor readings by training the SOM using the wavelet coefficients. Our proposed algorithm, the integrated SOM and DWT algorithm, could help reduce wasted energy caused by transmitting all measurement data to the base station by applying DWT algorithm onto the sensor modes in order to reduce size of transmitted data without losing the significant feature of the data.

II. ANOMALY DETECTION

The first step of anomaly detection involves selecting the data parameters to be monitored and grouping them together in a pattern vector $\mathbf{x}^{\mu} \in \Re$, $\mu = 1,\ldots,N$,

$$x^{\mu} = \begin{pmatrix} x_1^{\mu} \\ x_2^{\mu} \\ \vdots \\ x_n^{\mu} \end{pmatrix} = \begin{pmatrix} \mathrm{KPI}_1^{\mu} \\ \mathrm{KPI}_2^{\mu} \\ \vdots \\ \mathrm{KPI}_n^{\mu} \end{pmatrix}, \tag{1}$$

where μ is the observation index, n is the number of parameter types or key performance indices (KPIs) chosen to monitor the environmental condition.

A. Self-Organizing Map

Competitive neural models such as the self-organizing map (SOM) [13], are able to extract statistical regularities from the input data vectors and encode them in the weights without supervision. It maps a high-dimensional data manifold onto a low-dimensional, usually two-dimensional, grid or display.

The basic SOM consists of a regular grid of map units or neurons as shown in Fig 1(a). Each neuron, denoted by i (depicted by the black dot), has a set of layered neighboring neurons (depicted by the white dots) as shown in Fig 1(a).

Neuron i maintains a weight vector $\mathbf{m}_i$. In order to follow the properties of the input data, such vector is updated during the training process. For example, Fig.1(b) shows a SOM represented by a 2-dimensional grid of 4×4 neurons. The dimension of each vector is equal to the dimension of the input data. In the figure, a vector of input data (marked by x) is used to train the SOM weight vectors (the black dots). The winning neuron (marked by BMU) as well as its 1-neighborhood neurons, adjust their corresponding vectors to the new values (marked by the gray dots).

The SOM is trained iteratively. In each training step, one sample vector $\mathbf{x}$ from the input data set is chosen.

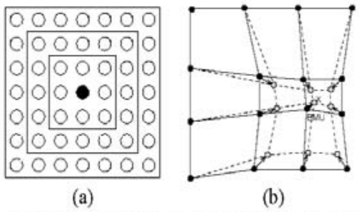


Fig. 1 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3- neighborhoods, (b) SOM updates the BMU with 1-neighborhood.

The distances between the sample data and all of weight vectors in the SOM are calculated using some distance measure. Suppose that at iteration t, neuron i whose weight vector $\mathbf{m}_i(t)$ is the closest to the input vector $\mathbf{x}(t)$. We denote such weight vector by $\mathbf{m}_c(t)$ and refer to it as the Best-Matching Unit (BMU), that is

$$\|x(t) - m_c(t)\| = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \tag{2}$$

where $\|\cdot\|$ is the Euclidian distance.

Suppose neuron i is to be updated, the SOM updating rule for the weight vector of neuron i is given by

$$m_i(t+1) = m_i(t) + \eta_t h_c(i,t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{3}$$

where t is the iteration index, $\mathbf{x}(t)$ is an input vector, η_t is the learning rate, $h_c(i,t)$ is the neighborhood function of the algorithm. The Gaussian neighborhood function may be used, that is

$$h_c(i,t) = \exp\left(-\frac{\|r_c(t) - r_i(t)\|^2}{2\sigma^2(t)} \right) \tag{4}$$

where $r_i(t)$ and $r_c(t)$ are the positions of neurons i and the BMU, c respectively, and $\sigma(t)$ is the radius of the neighborhood function at time t. Note that $h_c(i,t)$ defines the width of the neighborhood. It is necessary that $\lim_{t\to\infty} h_c(i,t) = 0$ and $\lim_{t\to\infty} \eta_t = 0$ for the algorithm to converge [13].

B. Discrete Wavelet Transform

DWT is a mathematical transform that separates the data signal into fine-scale information known as detail coefficients, and rough-scale information known as approximate coefficients. Its major advantage is the multi-resolution representation and time-frequency localization property for signals. Usually, the sketch of the original time series can be recovered using only the low-pass-cut off decomposition coefficients; the details can be modeled from the middle-level decomposition coefficients; the rest is usually regarded as noises or irregularities. The following equations describe the computation of the DWT decomposition process:

$$a_{j+1}^{DWT}(k) = \sum_n h_0(n-2k)a_j^{DWT}(k) \qquad (5)$$

$$d_{j+1}^{DWT}(k) = \sum_n g_0(n-2k)a_j^{DWT}(k)\ , \qquad (6)$$

where the rough-scale (or approximation) coefficients a_j^{DWT} are convolved separately with h_0 and g_0, the wavelet function and scaling function, respectively, *n* is the time scaling index, *k* is the frequency translation index for wavelet level *j*. The resulting coefficient is down-sampled by 2. This process splits a_j^{DWT} roughly in half, partitioning it into a set of fine-scale (or *detail)* coefficients d_{j+1}^{DWT} and a coarser set of approximation coefficients a_{j+1}^{DWT} [21].

DWT has the capability to encode the finer resolution of the original time series with its hierarchical coefficients. Furthermore, DWT can be computed efficiently in linear time, which is important while dealing with large datasets.

C. *Integration of SOM and DWT*

In the integration of SOM and DWT algorithm, the DWT algorithm is used as an input data preprocessor of the SOM algorithm in order to reduce the size of data without losing any significant feature of the data. This enables the implementation of in-network processing which helps to reduce the radio communication energy and eventually prolong the lifetime of the WSN [11]. The input data will be padded with zero if its length is odd data. After obtaining the wavelet coefficients, these coefficients will be fed to the SOM algorithm which can be divided into 2 sets. Each set contains both approximate and detail coefficients. The first set which is obtained from noiseless data, will be used to train the SOM algorithm. The second set which is obtained from the faulty data will be used to test the SOM algorithm. Then to reduce the false alarms the detected results will be double checked by using the univariate method [13], [14].

D. *Anomaly Detection*

A new observation data set can be considered abnormal if the distance between the weight vector of the winning neuron and the new state vector, given by

$$e^{\mu} = \left\| \mathbf{x}^{new} - \mathbf{m}_c^{\mu} \right\| \qquad (7)$$

is greater than a certain percentage $p = 1 - \alpha$ of the distances in the distance distribution profile. That is,

$$\begin{aligned} &\text{IF } \ e^{\mu} \in \left[e_p^-, e_p^+ \right], \\ &\text{THEN } \mathbf{x}^{new} \text{ is NORMAL} \qquad (8) \\ &\text{ELSE } \ \mathbf{x}^{new} \text{ is ABNORMAL.} \end{aligned}$$

Equation (8) is referred to as the global decision. In [6], an addition of local decisions of each KPIs is presented. Suppose that a data vector $\mathbf{x}^{new}$ is considered abnormal by the global decision. Then in the local anomaly detection, the absolute value of error in each component of the error vector is then computed by

$$\left| \mathbf{E}^{new} \right| = \begin{pmatrix} \left| x_1^{\mu} - m_{c,1}^{\mu} \right| \\ \left| x_2^{\mu} - m_{c,2}^{\mu} \right| \\ \vdots \\ \left| x_n^{\mu} - m_{c,n}^{\mu} \right| \end{pmatrix}. \qquad (9)$$

The error in each KPI is then compared to the interval of normality component-by-component, and the anomaly decision is carried out as in (8).

III. EXPERIMENT RESULTS

A. *Evaluation on detecting synthetic faults*

In this section, we evaluated the performance of the proposed integration of SOM and DWT algorithm by detecting anomalies in series of synthetic data and actual data collected from a wireless sensor network injected by various synthetic faults.

In the experiment, we generated the synthetic input data from a normal distribution N(0,1) and synthetic faults by additive white Gaussian noise (AWGN) with power 25 dBW generated from MATLAB. We used such fault because its statistical similarity to the synthetic input data thus, it is more difficult to be detected. Therefore, we can evaluate the performance of the algorithms under ambiguous faults. The amount of faults is represented by the notation n/s, where "n" is the amount of faults per series and "s" is the amount of series of faults, resulting in the total amount of n×s faults. The generated faults added to the input data ranged from bursty (20/10) to sparse (1/10). The exact positions of the faults injected in the input data were predetermined and was later used to detect true and false alarms. In the experiment using real data, we have chosen 2 parameters, namely temperature and moisture, as KPIs collected from samples of compost in a bioorganic fertilizer plant. In this paper, the data of the 2 KPIs at the WSNs were collected every 5 minutes for 3 days. We compared 3 anomaly detection methods: SOM algorithm, DWT algorithm, and integration of SOM and DWT algorithm.

We measured 2 performance metrics: 1) the *true alarm rate* which is defined by the number of detected true anomalies over the total number of true anomalies in the data set; and 2) the *false alarm rate* which is defined by the number of detected false alarms over the total number of detected anomalies.

In the DWT algorithm, we used the threshold in (11) in order to decide whether the data is normal or abnormal

$$\sigma_w = median(\left|d_1 - \overline{d_1}\right|) \tag{10}$$

$$T_w = \sigma_w \sqrt{2\log_e(N)}\,, \tag{11}$$

where N is the size of data and $\overline{d}_1$ is the sample mean of the level 1 detail coefficients [21].

This threshold was calculated from the low pass and high pass coefficients from the assumed normal data by using Haar and Daubechies4 mother wavelets. The Haar and Daubechies4 wavelets were used because they are relatively easy to cross-check by hand with computed coefficients from MATLAB program. Hence, we can compare the position of each coefficient with the actual fault position. After the threshold calculation, the set of coefficients which are obtained from the DWT of the noisy data will be compared with the threshold, coefficient by coefficient. For the real data scenario, the data was normalized by equation (12) before being processed by the DWT to eliminate potential outliers:

$$\text{Norm(Data)} = \frac{\text{(Data) - mean(Data)}}{\text{variance(Data)}} \tag{12}$$

If the absolute value of the coefficient is greater than the computed threshold, an anomaly is said to be detected.

In the SOM algorithm and the proposed integrated SOM and DWT algorithm, the initial value for learning rate in the SOM part was set to $\eta_0 = 0.9$, and gradually reduced to $\eta_T = 10^{-5}$, in order to guarantee convergence [13]. The number of training epochs was set to 50 because longer training epochs tend to over train the SOM [13]. The required percentage of distance in (8) was set to 99%. We used a Gaussian neighborhood function because the distribution of the collected data after the normalization fits well to the Gaussian distribution. The 30×30 size of neurons was used. Fig. 2 and 3 show that the anomaly detection in SOM algorithm and the integrated SOM and DWT algorithms improve as the number of neurons is increased. This suggests that the more neurons used, the "finer" SOM's classification becomes resulting in enhanced detection performance. However, at neuron size 50×50, the SOM requires much longer training time with a marginal improvement in the detection performance. Therefore, the 30×30 size of neurons was selected to train and test the SOM. We also improved the SOM algorithm by double checking with the univariate method in order to reduce the false alarm rate [13], [14]. To obtain accurate results, each algorithm was repeated for 70 runs.

To evaluate the performance of all algorithms, the results of each algorithm were compared to the (known) fault positions which were injected into the input data. In particular, when an anomaly was detected then its position was compared with the (known) fault position. If this position existed, then the anomaly detected was a true alarm; otherwise, it was a miss. On the other hand, if an anomaly was detected but the (known) fault position did not exist, then the anomaly was a false alarm.

Fig. 4 and Fig. 5 show the percentage of true alarm rate averaged over 70 runs, as a function of the amount of faults added into the input data. Note that the proposed integrated SOM and DWT algorithm which used Haar as a mother wavelet gives the best performance over other algorithms. This is because the DWT with Haar wavelet can detect changing points. In particular, the Haar wavelet uses 2 adjacent input data to compute a coefficient whereas the Daubechies4 uses 4 adjacent input data to compute a coefficient. However, Daubechies4 gave a lower performance than Haar because each coefficient was computed from an average over 4 input data. If a fault occurred in 1 of these 4 data, such fault will be averaged with the remaining 3 normal data resulting in a coefficient with an absolute value possibly lower than the decision threshold. Consequently, the true alarm rate is reduced. On the other hand, the Haar wavelet only uses 2 adjacent data to compute 1 coefficient. Thus, the true alarm rate is significantly higher than that of Daubechies4. The integrated SOM and DWT algorithm using Haar also outperforms the SOM algorithm. This is because in the Haar case, the coefficients obtained were transformed from two adjacent data. Therefore, if some data was faulty or differed greatly from the data nearby, this coefficient can detect such anomaly. On the other hand, the SOM algorithm directly checked the data one by one to detect an anomaly. If the data was faulty but had a small magnitude, then this fault may not be detected, and consequently the true alarm rate was reduced. Note that the DWT algorithm has the lowest performance because the decision threshold in (11) is rather conservative. Furthermore, the threshold is fixed throughout the detection and the algorithm does not have any double checking method.

Fig.4 and 5 show that the proposed algorithm can achieve up to 65% and 67% of true alarm rates in case of bursty faults for synthetic and real data, respectively. The proposed algorithm achieved a true alarm rate of up to 18% higher than the SOM algorithm alone in presence of bursty faults. Compared to the DWT alone, the proposed algorithm can attain a true alarm rate of up to 35% more in the bursty faults case.

As for sparse faults, the proposed algorithm can achieve up to 69% and 80% true alarm rates for synthetic and real data, respectively. The integrated SOM and DWT also gave true alarm rates of up to 10% higher than the SOM algorithm alone whereas DWT performed the weakest, in presence of sparse faults.

Fig.6 and Fig. 7 show the false alarm rate results in the synthetic and real data experiments, respectively. Note that most results have low false alarm rates, i.e., less than 1 % except in the case of sparse faults due to the increased detection difficulty.

The integration of SOM and Daubechies4 DWT also gave a weak performance due to the reasons previously explained. All these results show that the integration of SOM and DWT with Haar as a mother wavelet outperform the SOM algorithm and DWT method.

From these figures, the false alarm rate of the proposed algorithm is 0.11% and 0.13% in presence of bursty faults and 0.91% and 1% in presence of sparse faults with synthetic and real data, respectively. Note that the false alarm rate of the proposed algorithm is slightly higher than the other two algorithms. Since the gain in the true alarm rate is more significant, such tradeoff is therefore considered acceptable.

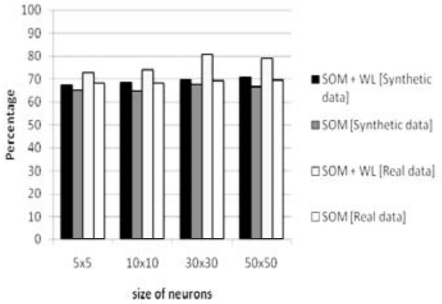


Fig. 2 True alarm rates with different size of neurons in the sparse faults case

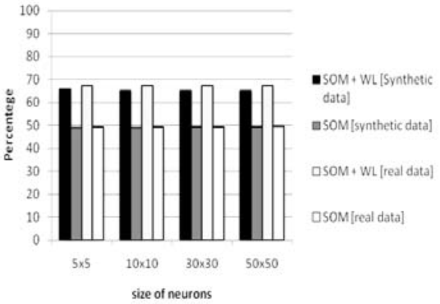


Fig. 3 True alarm rates with different size of neurons in the bursty faults case

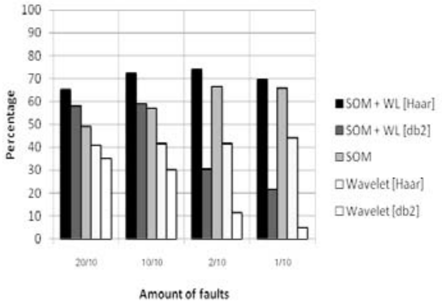


Fig. 4 True alarm rates with synthetic data

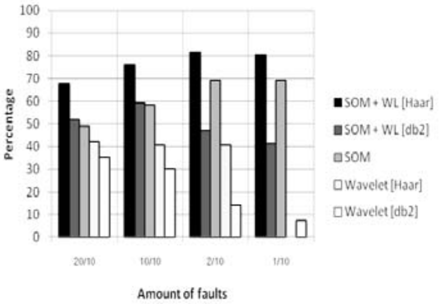


Fig. 5 True alarm rates with real data

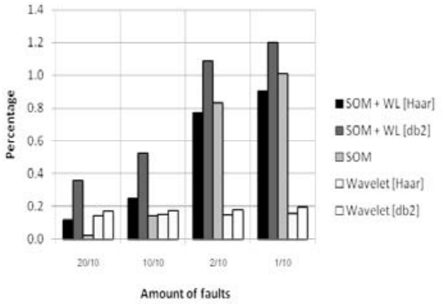


Fig. 6 False alarm rates with synthetic data

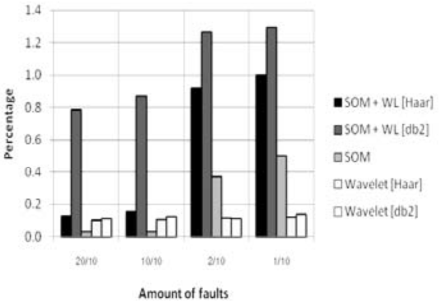


Fig. 7 False alarm rates with real data

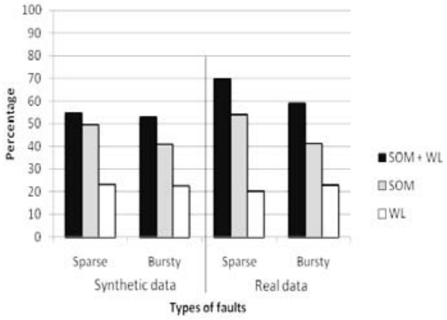


Fig. 8 True alarm rate with 10 dBW AWGN faults

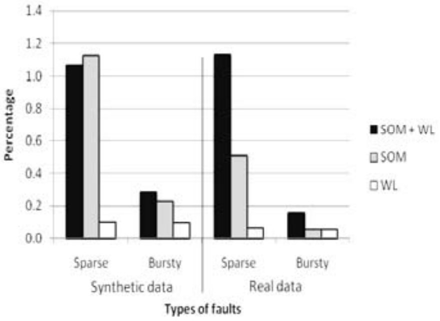


Fig. 9 False alarm rate with 10 dBW AWGN faults

Fig. 8 and 9 show the effect of the decreasing of AWGN noise power from 25dBW to 10dBW in both synthetic and real data scenarios. Only the Haar wavelet was used in the proposed algorithm and the DWT algorithm. The Daubechies4 was not included due to its weak performance. Though the anomaly detection is more difficult, the proposed integrated SOM and DWT still consistently outperforms the other two methods in terms of true alarm rate but with marginal increase in the false alarm rate as tradeoff.

The proposed integration of SOM and DWT algorithm with Haar wavelet outperformed the SOM algorithm and the DWT algorithm alone. Our results suggest that the proposed integrated SOM and DWT anomaly detection scheme can be deployed in a resource-constrained network such as a WSN. In particular, the DWT using Haar wavelet can be implemented at the sensor nodes as a data preprocessor to reduce the amount of data to be transmitted by at least half (for one-level DWT). Since energy consumption is critical in WSNs, such distributed in-network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN [11] while still maintaining acceptable anomaly detection accuracy.

B. Evaluation on detecting faults in real-world datasets

In this section, we apply the anomaly detection methods to three real-world datasets, i.e., NAMOS [8], INTEL Berkeley Lab [9], and SensorScope [10], to detect anomalies in sensor traces. However, since we did not have ground truth information about faults for these datasets, visual inspection and the histogram method are used to decide whether the data is normal or abnormal. The histogram method was used because it displays the data distribution which allows us to determine a suitable threshold according to that data series.

The histogram method divides the time series of sensor readings into groups of *N* samples. We then plot the histogram of the samples and select a threshold according to outliers of the histogram. However, this approach is sensitive to the choice of *N*. Fig. 10 [10] shows the effect of *N* on the histogram computed for sensor measurements taken from a real-world deployment [7]. Therefore, selecting the correct value for the parameter *N* requires a good understanding of the *normal* sensor readings. In practice, one should also try a range of values for *N* to ensure that the samples flagged as faulty are not just an artifact of the value selected for *N* [7]. With heuristic adjustments on the parameter value of N and some domain knowledge of the normal data profile, the histogram method was used as reference to identify abnormal data samples.

In the real-world datasets experiment, we evaluated the performance of 3 anomaly detection methods: the SOM, DWT using the Haar wavelet methods, and the integration of SOM and DWT using the Haar wavelet. For the SOM and the integration of SOM and DWT using Haar wavelet algorithms, we also considered the effects of changing the number of training samples, the number of training epochs which were 10 and 50 iterations, and the size of neurons which were 10x10 and 30x30. We also compared the performance of the low and high pass Haar wavelet coefficients (LP and HP, respectively) in the DWT algorithm and the integration of SOM and DWT algorithm.

1) NAMOS

In the NAMOS dataset, 9 buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, James Reserve for over 24 hours in August, 2006 [8]. We analyzed the measurements from chlorophyll sensors on buoys no. 103 for 10^4 samples as shown in Fig.11. In the experiment, the histogram method was used to identify anomalies in the NAMOS dataset from which we selected the threshold of 0 and 500 as lower and upper bounds of the normal region, respectively. The size of training samples of 1500 and 3000 samples were used to train both the SOM and the integration of SOM and DWT algorithms.

Fig. 12 shows the percentage of detection alarm rates for true, miss and false alarms which were obtained from changing the size of training samples. Note that both the SOM algorithm and the proposed integrated SOM and DWT algorithm with low pass wavelet coefficients gave the best true alarm detection performance of nearly 100% while their false alarm rates is negligible. The integrated SOM and DWT algorithm and DWT algorithm with high pass coefficients gave the lowest performance. This is because the high pass coefficients are more suitable for detecting the changing points of the data whereas most of faults appear constant as seen from $9x10^3$ samples onwards in Fig. 11. In addition, reducing the size of training samples did not have any effect on the anomaly detection in the SOM algorithm and the proposed integrated SOM and DWT algorithm. This is because both training samples are obtained from a normal period of data which differ only in sample sizes.

Fig. 13 shows the percentage of detection alarm rates for true, miss and false alarms which were obtained by reducing the number of training epoch from 50 to 10 iterations. In this case, the SOM algorithm gave the best performance with nearly 100% of true alarm detection rate and no false alarm rate. DWT algorithm which used low pass coefficient gave high performance while the proposed integrated SOM and DWT algorithm with either coefficient failed on detecting any anomaly. The reason could be caused by the constant features of the faults in NAMOS which may be difficult to decide whether samples are normal or abnormal, in particular, if the wavelet coefficients are under trained. Hence, care must be taken when selecting the suitable number of training epochs. In addition, we also investigated the effect of reducing the size of neurons. Results in Fig.14 show that there is no significant effect from reducing size of neurons from 30x30 to 10x10.

2) INTEL

In the INTEL dataset, 54 Mica2Dot motes with temperature, humidity and light sensors were deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004 [9]. In this paper, we present the results on the anomaly detection in the temperature readings.

In the experiment, we selected the threshold value of 16 and 30 as the upper and lower bounds of the normal data regions. These values were obtained from the histogram method. The size of training samples used was 1000 and 2000 samples as shown in Fig. 15.

Fig. 16, shows the percentage of detection alarm rates for true, miss and false alarms which were obtained from changing the size of training samples. According to the results as shown, the SOM and the proposed algorithm can achieve a true alarm rate of up to 100% with very small false alarm rate. Their true alarm rate is 67% higher than the DWT method using high pass coefficients. Note that the high pass coefficients can detect spike faults better than low pass coefficient since the high pass coefficients reflect the rate of change between two successive samples. Note that the DWT using low pass coefficient gave the lowest performance. The results of changing number of training epochs are shown in Fig. 17 and the size of neurons are shown in Fig. 18. From both figures there are no significant effects on the detection rate because the fault in this dataset has a high amplitude and can be easily detected.

3) *SensorScope*

The SensorScope project is an ongoing outdoor sensor network deployment consisting of weather-stations with sensors for sensing several environmental quantities such as temperature, humidity, solar radiation, soil moisture, and so on [10]. We did not have the ground truth regarding faulty samples for this dataset. We used a combination of visual inspection and the histogram method to identify anomaly samples [7].

In the experiment, we present the results on the anomaly detection in two types (KPIs) of data in the pdg2008-metro-1 dataset, i.e., the surface and ambient temperature readings. Using visual inspection and the histogram method, the lower and upper threshold values used for anomaly detection in SensorScope were -14 and 4 for the surface temperature and -12 and 4 for the ambient temperature. The sizes of training samples were 700 and 2000 samples for both KPIs as shown in Fig. 19.

Fig. 20 shows the percentage of detection alarm rates for true, miss and false alarms obtained from changing the size of training samples. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 2% higher than the SOM algorithm while false alarm rate remained less than 0.5%. The proposed algorithm using low pass coefficients can attain a true alarm rate of up to 17% more than the DWT algorithm alone. The integrated SOM and DWT algorithm and DWT algorithm which used high pass coefficients gave the lowest performance. This is because high pass coefficients are more suitable for short duration faults such as, spike or sparse faults while the data in Fig. 19 contains noise faults which affect a larger number of successive samples with an increase in their variance.

The effect of reducing the number of training epochs is shown in Fig. 21. According to the results, there is no significant effect on the performance of SOM and the integrated SOM and DWT.

Fig. 22 shows the percentage of detection alarm rates for true, miss and false alarms which were obtained from reducing the size of neurons. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 2% lower than the SOM algorithm, whereas the false alarm rate remains lower than 0.5%. On the other hand, the proposed algorithm using low pass coefficients can attain a true alarm rate of up to 13% more than the DWT algorithm alone.

The results from the real-world dataset show that our proposed algorithm, the integrated SOM and DWT algorithm performs as equally well as the SOM algorithm while using just half of the input data (using level 1 of DWT). This is because DWT is able to extract relevant data features without any significant loss in information, thereby reducing wasted energy from transmitting all measurements to the base station. Hence, by applying DWT onto the sensor modes, to achieve in-network data processing, the size of transmitted data can be reduced while still maintain good anomaly detection abilities.

However, a variety of data characteristics can affect the anomaly detection in the integrated SOM and DWT algorithm as can be seen from the NAMOS dataset. Hence, a suitable setting of the algorithm, such as the size of training epochs, has to be considered carefully. In terms of the number of neurons, the more neurons used, the finer SOM's classification becomes, generally resulting in enhanced detection performance. However, the results in the real-world datasets show that there is no significant change in detection performance. In terms of the selection of wavelet coefficients, high pass coefficients are more suitable for detecting the changing points of the data, whereas low pass coefficients are more suitable for detecting the changing of trend of the data. These settings can be determined by considering the nature of the sensors deployed. For example, calibration errors in sensors can cause offset faults (whereby the measured value can differ from the true value by a constant), low battery voltage causes a combination of noise and constant faults, while short faults are caused by software error during communication and data logging [7].

IV. CONCLUSION

This paper proposed an integration of a competitive learning method called the self-organizing map (SOM) and the discrete wavelet transform (DWT), to detect anomalies from synthetic faults and faults obtained from real-world datasets.

In the synthetic faults experiment, the results show that the integration of SOM and DWT with Haar as a mother wavelet can attain 65% and 67% of true alarm rates in the case of bursty faults, and 69% and 80% of true alarm rates in case of sparse faults for synthetic and real data, respectively. In terms of the true alarm rate, the proposed algorithm outperforms the SOM algorithm by up to 18% and DWT algorithm by up to 35% in presence of bursty faults. With sparse faults, the proposed algorithm can gain a true alarm rate up to 10% above the SOM algorithm alone and entirely outperforms the DWT algorithm alone. Such gain in true alarm rates come with a marginal increase of false alarm rate.

In the real-world datasets, the integration of SOM and DWT with Haar as a mother wavelet can attain up to 99%, 100% and 83% of true alarm rates in the NAMOS, INTEL and SensorScope dataset, respectively. Our proposed algorithm also performs as equally well as the SOM algorithm and outperforms the DWT algorithm by up to 15%, 100%, and 17% in the NAMOS, INTEL and SensorScope dataset, respectively.

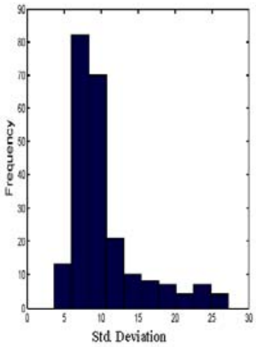


Fig. 10a Histogram shape with N = 100

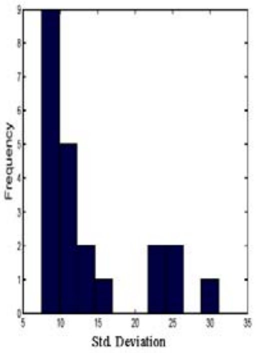


Fig. 10b Histogram shape with N = 1000

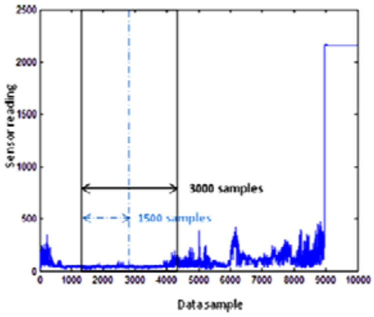


Fig. 11 NAMOS dataset of 10^4 samples

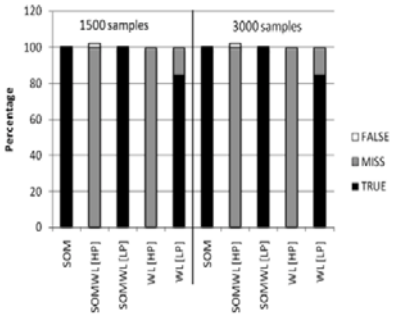


Fig. 12 Detection rate in the NAMOS dataset using training epoch of 50 iterations

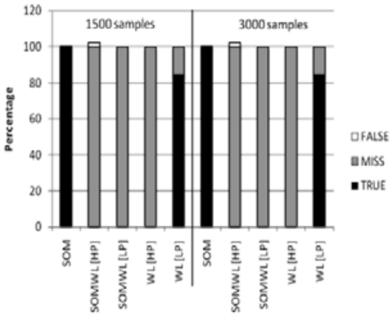


Fig. 13 Detection rate in the NAMOS data set using training epoch of 10 iterations

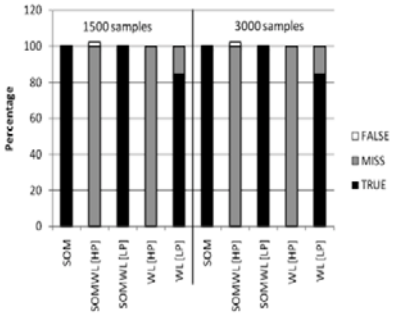


Fig. 14 Detection rate in the NAMOS dataset using the size of neurons of 10x10

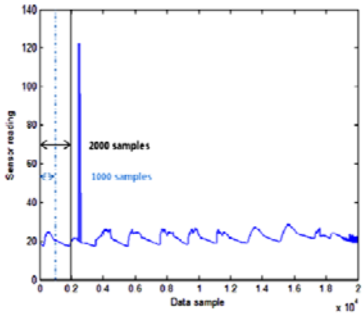


Fig. 15 INTEL dataset of $2x10^4$ samples

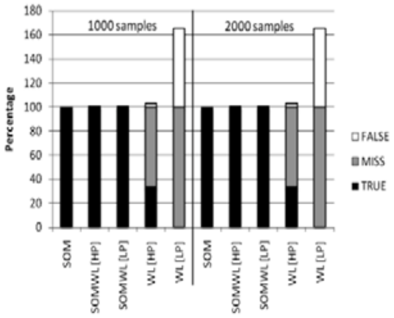


Fig. 16 Detection rate in the INTEL dataset using training epoch of 50 iterations

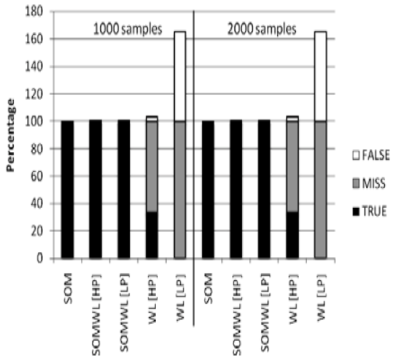


Fig. 17 Detection rate in the INTEL dataset using training epoch of 10 iterations

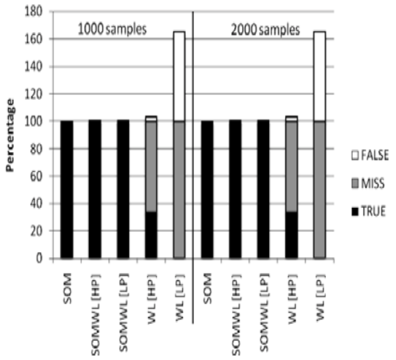


Fig. 18 Detection rate in the INTEL dataset using the size of neurons of 10x10

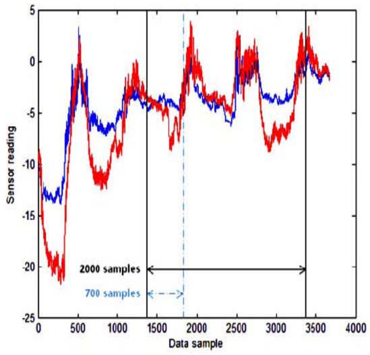


Fig. 19 SensorScope dataset of 4000 samples

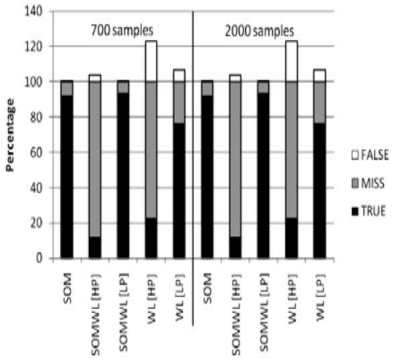


Fig. 20 Detection rate in the SensorScope dataset using training epoch of 50 iterations

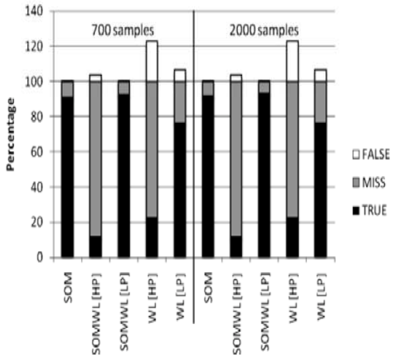


Fig. 21 Detection rate in the SensorScope dataset using training epoch of 10 iterations

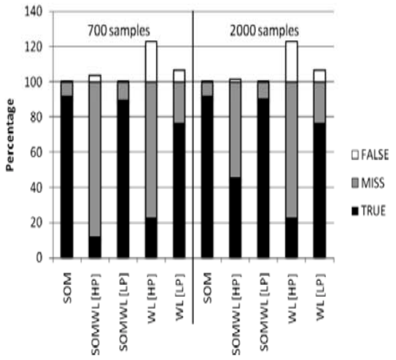


Fig. 22 Detection rate in the SensorScope dataset using the size of neurons of 10x10

In terms of the true alarm rate when reducing the number of training epochs, the proposed algorithm has a poor performance due to the detection ability of wavelet coefficients is unsuitable for the anomaly in the NAMOS dataset. In the INTEL dataset, the proposed algorithm outperforms the DWT algorithm and performs equally well when compared to the SOM algorithm while using just half of the input data. In the SensorScope dataset, the proposed algorithm outperforms the DWT algorithm but is slightly lower than the SOM algorithm.

By reducing the size of neurons, the proposed algorithm still obtained a true alarm rate up to 16%, 100% and 84% higher than the DWT algorithm in NAMOS, INTEL and SensorScope dataset, respectively. The proposed algorithm performed equally well as the SOM algorithm in the NAMOS and INTEL dataset but only 2% lower than the SOM algorithm in the SensorScope dataset. The reduction of the size of neurons did not show any significant change in detection performance.

Our results suggest that the integration of SOM and DWT with Haar wavelet can lead to more effective anomaly detection. In particular, our results confirm that the proposed algorithm can maintain acceptable anomaly detection accuracy while using just half of the input data (using level 1 DWT).

In the future, we plan to extend our work to investigate anomaly detection with actual faults obtained from the bioorganic fertilizer plant environment, and study its performance by increasing the DWT level and considering other different types of wavelets. Furthermore, we also plan to investigate ways to identify and eliminate erroneous sensor readings at the sensor nodes, which could help further reduce wasted energy from transmitting unwanted erroneous measurements to the base station.

V. ACKNOWLEDGMENT

The authors would like to thank Assoc. Prof. Dr. Boon Hee Soong and Mr. Edwin Chan of Intelligent Systems Laboratory, Nanyang Technological University, Singapore for their assistance and suggestions on the sensor development and implementation.

REFERENCES

[1] G. Kaur, V. Saxena and J.P. Gupta, “Anomaly Detection in Network Traffic and Role of Wavelets,” *IEEE Trans. on Instrumentation and Measurement*, vol. 7, no. 5, pp. 46-51, April. 2010.

[2] D.E. Dening, “An Intrusion-Detection Model,” *IEEE Trans. on Software Engineering*, vol. SE-13, vo. 2, pp. 222-232, 1987.

[3] S. Pervez, I. Ahmad, A. Akram and S.U. Swati, “A Comparative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems,” *WSEAS Int. Conf. on Multimedia, Internet & Video Technol*ogies, pp. 84-89, 2006.

[4] M. Thottan and J. Chuanyi, “Anomaly detection in IP network,” *IEEE Trans. on Signal Processing*, vol.51, no.8, 2003.

[5] J. Laiho, K. Raivio, P. Lehtimaki, K. Hatonen, and O. Simula, “Advanced Analysis Methods for 3G Cellular Networks,” *IEEE Trans. on Neural Networks*, vol.4, no.3, pp. 930-942, 2005.

[6] M.H. Lee and Y.H. Choi, “Fault detection of wireless sensor networks,” *Computer Communications*, vol. 31, pp. 3469-3475, 2008.

[7] A.B. Sharma, L. Golubchik, and R. Govindan, “Sensor Faults: Detection Methods and Prevalence in Real-World Datasets,” *Trans. on Sensor Networks*, vol.5, pp. 1-34, 2010.

[8] NAMOS. 2006. Networked Aquatic Microbial Observing System. Data set available at: http://robotics.usc.edu/~namos/data/jr aug 06/.

[9] INTEL. 2004. The Intel Lab Data. Data set available at: http://berkeley.intel-research.net/labdata/.

[10] SENSORSCOPE. 2006. The SensorScope Lausanne Urban Canopy Experiment (LUCE) Project. Data set available at: http://sensorscope.epfl.ch/index.php/LUCE.

[11] S. Rajasegarar, C. Leckie, and Palaniswami, “Anomaly Detection in Wireless Sensor Networks,” *IEEE Wireless Communications*, vol.15, no.4, pp. 34-40, 2008.

[12] J. Laiho, M. Kylvaja, and A. Hoglund, “Utilization of advanced analysis methods in UMTS networks,” *IEEE Vehicular Technology Conf.*, vol. 2, pp. 726-730, May. 2002.

[13] G.A. Barreto, J.C. Mota, L.G. Souza, R.A. Frota, and L. Aguaya, “Condition monitoring of 3G cellular network through,” *IEEE Trans. Neural Networks*, vol. 16, no. 5, pp. 1064-1075, Sep. 2006.

[14] P. Sukkhawatchani and W. Usaha, “Performance Evaluation of Anomaly Detection in Cellular Core Networks using Self-Organizing Map,” *Proc. of ECTI-CON 2008*, vol.1, pp. 361-364, May. 2008.

[15] J. Zheng and M. Hu, “Detection of TCP Attacks Using SOM with Fast Nearest-Neighbor Search,” *WSEAS Int. Conf. on Neural Networks*, pp.176-182, 2005.

[16] L. Paladina, M. Paone, G. Jellamo, and A. Puliafito, “Self organizing maps for distributed localization in wireless sensor networks,” *Computers and Communications, 2007,* 12th IEEE Symposiumpp, pp.1113-1118, July. 2007.

[17] V.A. Aquino and J.A. Barria, “Anomaly detection in communication Networks using wavelets,” *IEEE Proc. in Commun*ications, vol.148, no.6, pp. 355-362, Dec. 2001.

[18] N. Yadaiah, and Nagireddy Ravi, “Fault detection techniques for power transformers,” *Industrial & Commercial Power Systems Technical Conf.*, pp. 1- 9, 2007.

[19] Z. Xu and Q. Zhao, “A novel approach to fault detection and isolation based on wavelet analysis and neural network,” *Electrical and Computer Engineering*, vol. 1, pp. 572–577, May. 2002.

[20] S. Postalcıoglu, K. Erkan and ED. Bolat, “Implementation of Intelligent Active Fault Tolerant Control System,” *Springer-Verlag Berlin Heidelberg 2007*, pp. 804–812.

[21] R. J. Brychta, S. Tuntrakool, M. Appalsamy and D. Robertson, “Wavelet Methods for Spike Detection in Mouse Renal Synpathetic Nerve Activity,” *IEEE Trans. Biomedical Engineering*, vol.54, no.1, pp. 82-93, Jan. 2007.

BIOGRAPHY

Mr. Supakit Siripanadorn was born on September 21, 1984 in Nakhon Phanom province, Thailand. He finished high school education from Patumtepwittayakarn School, Nongkhai province. He received his Bachelor’s Degree in Engineering (Telecommunication) from Suranaree University of Technology in 2007. For his post-graduate, he continued to study with a Master’s degree in the Telecommunication Engineering Program, Institute of Engineering, Suranaree University of Technology. During Master’s degree education, he was a visiting researcher at IntelliSys, Nanyang Technological University, Singapore in a topic of wireless sensor development.