DATA COMPRESSION AND ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS

Saowaluk Takianngam

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 2011

การบีบอัดข้อมูลและการตรวจจับความผิดปกติในเครือข่ายตัวตรวจรู้ไร้สาย

นางสาวเสาวลักษณ์ ตะเคียนงาม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2554

DATA COMPRESSION AND ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS

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)

(Asst. Prof. Dr.Paramate Horkaew)

Member

(Prof. Dr.Sukit Limpijumnong)

Vice Rector for Academic Affairs

(Assoc. Prof. Flt. Lt. Dr.Kontorn Chamniprasart)

Dean of Institute of Engineering

เสาวลักษณ์ ตะเคียนงาม : การบีบอัดข้อมูลและการตรวจจับความผิดปกติในเครือข่ายตัวตรวจรู้ไร้สาย (DATA COMPRESSION AND ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. วิภาวี หัตถกรรม, 179 หน้า.

เครือข่ายตัวตรวจรู้ไร้สายมีข้อจำกัดหลายอย่าง เช่น หน่วยความจำ, ความกว้างแถบความถี่ (แบนด์วิธ), อัตราการส่งข้อมูลต่ำ, แหล่งพลังงานและการใช้พลังงาน, และความสามารถเชิงประมวลผล เป็นต้น ข้อจำกัดของอุปกรณ์เหล่านี้ส่งผลกระทบต่อความสามารถในการตรวจจับความผิดปกติของตัวตรวจรู้ และสามารถก่อให้เกิดความเสียหายต่อผลผลิตได้ นอกจากนี้ข้อจำกัดด้านแหล่งพลังงานในเครือข่ายตัวตรวจรู้ไร้สาย ต้องการใช้พลังงานให้น้อยที่สุด เนื่องจากการส่งข้อมูลในเครือข่ายตัวตรวจรู้ไร้สายนั้น ใช้พลังงานมากกว่ากระบวนการในการประมวลผล และการคำนวณข้อมูลที่มีขนาดเล็กจะใช้พลังงานน้อยกว่าการคำนวณข้อมูลขนาดใหญ่ด้วย

ดังนั้น งานวิจัยนี้จึงมุ่งเน้นที่จะผสมผสานการทำงานระหว่าง การบีบอัดข้อมูลด้วย Discrete Wavelet Transform (DWT) และ Lifting Wavelet Transform (LWT) ร่วมกับ การตรวจจับความผิดปกติของข้อมูลโดยใช้ One-Class Support Vector Machine (OCSVM)

วิธีการแรกที่นำเสนอในงานวิจัยนี้ คือ OCSVM + DWT เมื่อนำไปเปรียบเทียบกับวิธีการก่อนหน้านี้ที่ถูกนำเสนอมาแล้ว เช่น Self-Organizing Map (SOM) + DWT พบว่า OCSVM + DWT สามารถเพิ่มประสิทธิภาพในการตรวจจับความผิดปกติได้ สำหรับการทดลองกับข้อมูลสังเคราะห์พบว่า OCSVM + DWT ที่เลือกใช้ค่าสัมประสิทธิ์ความถี่ต่ำ มีอัตราความถูกต้องในการตรวจจับความผิดปกติถึง 100% ในขณะที่อัตราความผิดพลาดในการตรวจจับข้อมูลเพิ่มขึ้นเพียงเล็กน้อย เมื่อเปรียบเทียบกับวิธีการอื่น ๆ และในการทดลองกับชุดข้อมูลจริงพบว่า OCSVM + DWT ที่เลือกใช้ค่าสัมประสิทธิ์ความถี่ต่ำ ทำงานได้ดีที่สุด โดยมีอัตราความถูกต้องในการตรวจจับความผิดปกติสูงถึงเกือบ 100% แม้ว่าในการทดลองกับข้อมูลที่มีความผิดปกติแบบ Short หรือ Noise จะให้อัตราการตรวจจับข้อมูลที่ผิดพลาดสูงกว่าวิธีการอื่น ๆ ก็ตาม จากการทดลองจะเห็นว่า OCSVM + DWT ที่เลือกใช้ค่าสัมประสิทธิ์ความถี่ต่ำ เหมาะกับการตรวจจับข้อมูลที่มีความผิดปกติแบบ Short หรือ Noise เป็นองค์ประกอบ ในขณะที่ SOM + DWT ที่เลือกใช้ค่าสัมประสิทธิ์ความถี่ต่ำ เหมาะกับการตรวจจับข้อมูลที่มีความผิดปกติแบบ Constant เป็นองค์ประกอบ

อีกวิธีการหนึ่งที่นำเสนอในงานวิจัยนี้ คือ OCSVM + LWT ซึ่งจะถูกนำไปเปรียบเทียบประสิทธิภาพการทำงานกับวิธีการอื่น ซึ่งได้แก่ OCSVM + DWT และ OCSVM + Principal

Component Analysis (PCA) สำหรับการทดลองกับข้อมูลสังเคราะห์และข้อมูลจริงที่มีความผิดปกติแบบ Short เป็นองค์ประกอบ พบว่า OCSVM + LWT มีประสิทธิภาพการทำงานใกล้เคียงกับ OCSVM, OCSVM + DWT และ OCSVM + PCA สำหรับการทดลองกับข้อมูลสังเคราะห์และข้อมูลจริงที่มีความผิดปกติแบบ Noise และ Constant เป็นองค์ประกอบ พบว่า OCSVM + LWT และ OCSVM + DWT ที่เลือกใช้สัมประสิทธิ์ความถี่ต่ำ มีประสิทธิภาพการทำงานที่ดีกว่า OCSVM และ OCSVM + PCA ในทางกลับกัน OCSVM + LWT และ OCSVM + DWT ที่เลือกใช้สัมประสิทธิ์ความถี่สูง มีประสิทธิภาพการทำงานที่แย่ที่สุด ซึ่งแสดงให้เห็นว่า LWT มีความต้องการที่น้อยกว่า DWT ในแง่ของหน่วยความจำที่ใช้งาน และเวลาในการคำนวณ และจากผลการทดลองของเราแสดงให้เห็นว่า OCSVM + LWT เหมาะที่จะนำไปติดตั้งเครือข่ายตัวตรวจรู้ไร้สายมากกว่าวิธีการอื่น ๆ ที่ได้กล่าวมาแล้ว

สาขาวิชาวิศวกรรมโทรคมนาคม ลายมือชื่อนักศึกษา _____

ปีการศึกษา 2554 ลายมือชื่ออาจารย์ที่ปรึกษา _____

SAOWALUK TAKIANNGAM : DATA COMPRESSION AND ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS. THESIS ADVISOR : ASST. PROF. WIPAWEE HATTAGAM, Ph.D., 179 PP.

DATA COMPRESSION / ANOMALY DETECTION / WIRELESS SENSOR NETWORKS / ONE-CLASS SUPPORT VECTOR MACHINES (OCSVM) / SELF-ORGANIZING MAP (SOM) / LIFTING WAVELET TRANSFORM (LWT) / DISCRETE WAVELET TRANSFORM (DWT) / PRINCIPAL COMPONENT ANALYSIS (PCA)

Wireless sensor networks (WSNs) have many limitations such as memory, bandwidth, low-rate radio communication, energy supply and consumption, and computational capabilities. These limitations can affect the sensor node ability to detect anomalies and can damage produce. Furthermore, the battery supply limitations in WSNs require minimal energy consumption. Since radio communication in WSNs consume more energy than processing and computing, computation with small datasets is likely to consume less energy than a large dataset.

Therefore, this research is focused on incorporating the discrete wavelet transform (DWT) and lifting wavelet transform (LWT) data compression schemes with one-class support vector machine (OCSVM) anomaly detection.

Our first proposed algorithm (OCSVM + DWT) was compared with a previous algorithm i.e., self-organizing map (SOM) + DWT. We found the OCSVM + DWT can increase the efficiency of anomaly detection. For synthetic data, the OCSVM + DWT with low-pass coefficients (LP) achieved 100% detection rate (DR)

with marginal increase in false positive rate (FPR) when compared with all other algorithms. For real world datasets, the OCSVM + DWT with LP coefficients performed best by achieving nearly 100% DR although with slightly higher FPR for datasets containing short and noise faults. These results suggest that OCSVM + DWT (LP) algorithm is suited for short and noise faults whereas SOM + DWT (LP) is suited for short and constant faults.

Our second proposed algorithm (OCSVM + LWT) was compared with other variants of integration such as OCSVM + DWT and OCSVM + principal component analysis (PCA) and OCSVM alone (with uncompressed data). For synthetic data and real world datasets with short faults, the OCSVM + LWT performed equally well as the OCSVM alone, OCSVM + DWT and OCSVM + PCA. For synthetic data and real world datasets with noise and constant faults, the OCSVM + LWT [LP] and the OCSVM + DWT [LP] gave better performance than the OCSVM alone and OCSVM + PCA. On the contrary the OCSVM + LWT [HP] with high-pass coefficients and the OCSVM + DWT [HP] gave the worst performance. It was also demonstrated that LWT was less demanding in terms of memory requirement and computation time than DWT. Our results therefore suggest that OCSVM + LWT was more suitable for implementation in WSNs.

School of Telecommunication Engineering Student's Signature _____

Academic Year 2011 Advisor's Signature _____

ACKNOWLEDGEMENT

I would like to express my sincere thanks to my thesis advisor, Asst. Prof. Dr. Wipawee Hattagam for her invaluable help and constant encouragement throughout the course of this research. I am most grateful for her 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 her.

In addition, I am grateful for Assoc. Prof. Dr. Kitti Attakitmongcol, Asst. Prof. Capt. Dr. Prayoth Kumsawat for their comments and insights, and other faculty members in the School of Telecommunication Engineering for their suggestion and all their help.

I would also like to thank Asst. Prof. Dr. Peerapong Uthansakul, Asst. Prof. Dr. Paramate Horkaew for accepting to serve in my committee.

My sincere appreciation goes to Ms. Alisa Srikram and Ms. Pranitta Arthans 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 master's and doctoral degree courses for all their support throughout the period of this research.

Saowaluk Takianngam

TABLE OF CONTENTS

Page

ABSTRACT (THAI) I

ABSTRACT (ENGLISH) III

ACKNOWLEDGEMENT V

TABLE OF CONTENTS VI

LIST OF TABLE XI

LIST OF FIGURE XII

SYMBOLS AND ABBREVIATIONS XXIII

CHAPTER

I INTRODUCTION 1

1.1 Background problem and significance of study 1

1.1.1 Anomaly detection techniques 2

1.1.1.1 The parametric statistical anomaly detection techniques 3

1.1.1.2 Nonparametric anomaly detection techniques 4

1.1.2 Data compression techniques 10

1.2 Research objectives 13

1.3 Research hypothesis 14

1.4 Basic agreements 14

TABLE OF CONTENTS (Continued)

Page

1.5 Scope and limitation 14

1.6 Expected benefit 15

1.7 Synopsis of Thesis 15

II BACKGROUND THEORY 17

2.1 Anomaly detection 17

2.1.1 Self-organizing map (SOM) 19

2.1.2 One-class support vector machines (OCSVM) 22

2.2 Data compression 26

2.2.1 Principal component analysis (PCA) 27

2.2.2 Discrete wavelet transforms (DWT) 29

2.2.3 Lifting wavelet transforms (LWT) 31

2.3 Summary 33

III DISCRETE WAVELET TRANSFORM AND ONE-CLASS SUPPORT VECTOR MACHINES FOR ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS 35

3.1 Introduction 36

3.2 Anomaly detection 37

3.2.1 One-class support vector machines (OCSVM) 38

3.2.2 Self-organizing map (SOM) 41

3.3 Data compression 43

TABLE OF CONTENTS (Continued)

Page

3.3.1 Discrete wavelet transform (DWT) 43

3.4 Datasets for experiment 44

3.4.1 Synthetic data 46

3.4.2 INTEL dataset 46

3.4.3 SensorScope station no.39 dataset (SS39) 47

3.4.4 SensorScope pdg2008-metro-1 dataset (pdg2008) 47

3.4.5 NAMOS dataset 47

3.5 Experiment results 48

3.5.1 Evaluation of DWT with OCSVM 48

3.5.2 Comparison with previous work 54

3.6 Conclusion 60

IV LIFTING WAVELET TRANSFORM AND ONE-CLASS SUPPORT VECTOR MACHINES FOR ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS 62

4.1 Introduction 62

4.2 Anomaly detection 64

4.2.1 One-class support vector machines (OCSVM) 65

4.3 Data compression 68

4.3.1 Principal component analysis (PCA) 68

4.3.2 Discrete wavelet transform (DWT) 69

TABLE OF CONTENTS (Continued)

Page

4.3.3 Lifting wavelet transform (LWT) 70

4.4 Experiment results 72

4.4.1 Datasets for experiment 72

4.4.1.1 Synthetic data 73

4.4.1.2 Real world datasets 77

4.4.2 Performance evaluation 83

4.4.3 Extending to 3 KPI datasets 88

4.4.4 Computation time evaluation 93

4.5 Conclusion 94

V CONCLUSION AND FUTURE WORK 95

5.1 Conclusion 95

5.1.1 Anomaly detection DWT coefficients 95

5.1.2 Anomaly detection LWT coefficients 96

5.2 Future work 96

5.2.1 Increasing DWT and LWT level 96

5.2.2 Exploring other types of wavelets 97

5.2.3 Implementation on the sensor nodes 97

5.2.4 Comparison with other data compression techniques 97

5.2.5 Enhancing to fault predictability 97

TABLE OF CONTENTS (Continued)

Page

5.2.6 Normalize data prior feeding to anomaly detection 98

REFERENCES 99

APPENDICES

APPENDIX A DATASETS FOR EXPERIMENT CHAPTER 3 107

1. Synthetic data 108
2. INTEL dataset 114
3. SensorScope Station no.39 dataset (SS39) 116
4. SensorScope pdg2008-metro-1 dataset (pdg2008) 118
5. NAMOS dataset 122

APPENDIX B DATASETS FOR EXPERIMENT CHAPTER 4 125

1. Synthetic data 126
2. INTEL dataset 141
3. SensorScope pdg2008-metro-1 dataset (pdg2008) 150
4. NAMOS dataset 159

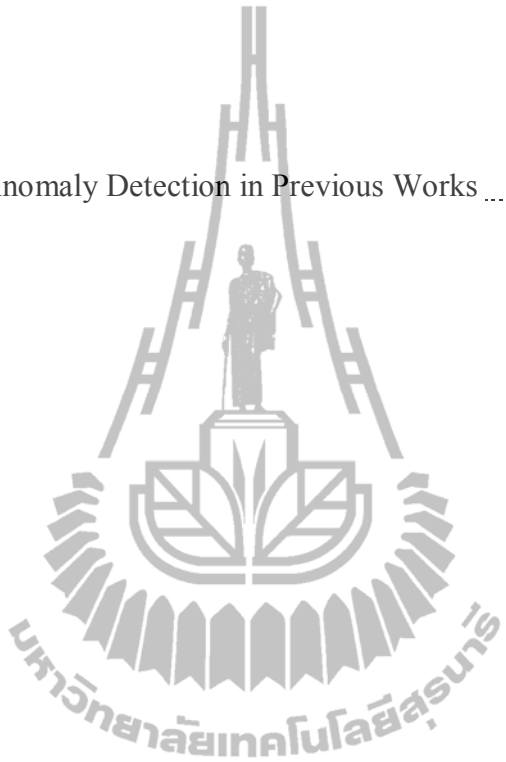
APPENDIX C NORMALIZATION EFFECT 168

APPENDIX D PUBLICATION 172

BIOGRAPHY 180

LIST OF TABLES

Table		Page
1.1	Ability of Anomaly Detection in Previous Works	9



LIST OF FIGURES

Figure **Page**

2.1 An illustration of the SOM 19

2.2 Geometry of the quarter-sphere OCSVM 25

2.3 Principal component analysis (PCA) framework 27

2.4 Discrete wavelet transforms (DWT) framework 30

2.5 Lifting Wavelet Transform (LWT) framework 31

3.1 Geometry of the quarter-sphere OCSVM 40

3.2 An illustration of the SOM 41

3.3 Fault in sensor readings 45

3.4 ROC for synthetic data inject 1/80 fault 49

3.5 ROC for synthetic data inject 5/16 fault 49

3.6 ROC for synthetic data inject 10/8 fault 50

3.7 ROC for synthetic data inject 20/4 fault 50

3.8 ROC for INTEL dataset 51

3.9 ROC for SS39 dataset 51

3.10 ROC for pdg2008-metro-1 dataset 52

3.11 ROC for NAMOS dataset 52

3.12 Detection Rate with different algorithm for 1/80 fault synthetic data 56

3.13 Detection Rate with different algorithm for 5/16 fault synthetic data 57

LIST OF FIGURES (Continued)

Figure **Page**

3.14 Detection Rate with different algorithm for 10/8 fault synthetic data 57

3.15 Detection Rate with different algorithm for 20/4 fault synthetic data 58

3.16 Detection Rate with different algorithm for the INTEL dataset 58

3.17 Detection Rate with different algorithm for the SS39 dataset 59

3.18 Detection Rate with different algorithm for the pdg2008 dataset 59

3.19 Detection Rate with different algorithm for the NAMOS dataset 60

4.1 Geometry of the quarter-sphere OCSVM 64

4.2 Principal Component Analysis (PCA) frameworks 69

4.3. Lifting Wavelet Transform (LWT) frameworks 71

4.4 Faults in sensor readings 73

4.5 2KPI Synthetic data with 1/80 faults 74

4.6 3KPI Synthetic data with 1/80 faults 75

4.7 2KPI Synthetic data with 20/4 faults 75

4.8 3KPI Synthetic data with 20/4 faults 76

4.9 2KPI Synthetic data with 80/1 fault 76

4.10 3KPI Synthetic data with 80/1 fault 77

4.11 INTEL dataset (temperature reading) 78

4.12 INTEL dataset (humidity reading) 78

4.13 INTEL dataset (voltage reading) 79

4.14 NAMOS dataset (fluorimeter reading) 80

LIST OF FIGURES (Continued)

Figure **Page**

4.15 NAMOS dataset (temperature reading from sensor 1) 80

4.16 NAMOS dataset (temperature reading from sensor 2) 81

4.17 pdg2008 dataset (ambient temperature reading) 82

4.18 pdg2008 dataset (surface temperature reading) 82

4.19 pdg2008 dataset (solar radiation reading) 83

4.20 ROC curve for 2KPI Synthetic data injected with 1/80 faults 85

4.21 ROC curve for 1KPI INTEL dataset (containing short faults) 85

4.22 ROC curve for 2KPI Synthetic data injected with 20/4 faults 86

4.23 ROC curve for 2KPI pdg2008 dataset (containing noise faults) 86

4.24 ROC curve for 2KPI Synthetic data injected with 80/1 fault 87

4.25 ROC curve for 1KPI NAMOS dataset (containing constant fault) 87

4.26 ROC curve for 3KPI Synthetic data injected with 1/80 faults 90

4.27 ROC curve for 3KPI INTEL dataset (containing short fault) 90

4.28 ROC curve for 3KPI Synthetic data injected with 20/4 faults 91

4.29 ROC curve for 3KPI pdg2008 dataset (containing noise fault) 91

4.30 ROC curve for 3KPI Synthetic data injected with 80/1 fault 92

4.31 ROC curve for 3KPI NAMOS dataset (containing constant fault) 92

4.32 Computation time of each data compression technique 93

A.1 Synthetic data without fault for training the SOM algorithm 108

A.2 Synthetic data with 1/80 faults 108

LIST OF FIGURES (Continued)

Figure **Page**

A.3 DWT low-pass coefficient of synthetic data with 1/80 faults 109

A.4 DWT high-pass coefficient of synthetic data with 1/80 faults 109

A.5 Synthetic data with 5/16 faults 110

A.6 DWT low-pass coefficient of synthetic data with 5/16 faults 110

A.7 DWT high-pass coefficient of synthetic data with 5/16 faults 111

A.8 Synthetic data with 10/8 faults 111

A.9 DWT low-pass coefficient of synthetic data with 10/8 faults 112

A.10 DWT high-pass coefficient of synthetic data with 10/8 faults 112

A.11 Synthetic data with 20/4 faults 113

A.12 DWT low-pass coefficient of synthetic data with 20/4 faults 113

A.13 DWT high-pass coefficient of synthetic data with 20/4 faults 114

A.14 Histogram of INTEL dataset (temperature reading) 114

A.15 INTEL dataset (temperature reading) 115

A.16 DWT low-pass coefficient of INTEL dataset (temperature reading) 115

A.17 DWT high-pass coefficient of INTEL dataset (temperature reading) 116

A.18 Histogram of SensorScope Station no.39 (SS39) dataset (global current reading) 116

A.19 SensorScope Station no.39 (SS39) dataset (global current reading) 117

A.20 DWT low-pass coefficient of SensorScope Station no.39 (SS39) dataset (global current reading) 117

LIST OF FIGURES (Continued)

Figure **Page**

A.21 DWT high-pass coefficient of SensorScope Station no.39 (SS39) dataset (global current reading) 118

A.22 Histogram of pdg2008 dataset (surface temperature reading) 118

A.23 pdg2008 dataset (ambient temperature reading) 119

A.24 pdg2008 dataset (surface temperature reading) 119

A.25 DWT low-pass coefficient of pdg2008 dataset (surface temperature reading) 120

A.26 DWT high-pass coefficient of pdg2008 dataset (surface temperature reading) 120

A.27 pdg2008 dataset (ambient temperature reading) 121

A.28 DWT low-pass coefficient of pdg2008 dataset (ambient temperature reading) 121

A.29 DWT high-pass coefficient of pdg2008 dataset (ambient temperature reading) 122

A.30 Histogram of NAMOS dataset (fluorimeters reading) 122

A.31 NAMOS dataset (fluorimeters reading) 123

A.32 DWT low-pass coefficient of NAMOS dataset (fluorimeters reading) 123

A.33 DWT high-pass coefficient of NAMOS dataset (fluorimeters reading) 124

B.1 2KPI Synthetic data with 1/80 faults 126

B.2 DWT low-pass coefficient of 2KPI synthetic data with 1/80 faults 126

LIST OF FIGURES (Continued)

Figure **Page**

B.3 DWT high-pass coefficient of 2KPI synthetic data with 1/80 faults 127
B.4 LWT low-pass coefficient of 2KPI synthetic data with 1/80 faults 127
B.5 LWT high-pass coefficient of 2KPI synthetic data with 1/80 faults 128
B.6 3KPI Synthetic data with 1/80 faults 128
B.7 DWT low-pass coefficient of 3KPI synthetic data with 1/80 faults 129
B.8 DWT high-pass coefficient of 3KPI synthetic data with 1/80 faults 129
B.9 LWT low-pass coefficient of 3KPI synthetic data with 1/80 faults 130
B.10 LWT high-pass coefficient of 3KPI synthetic data with 1/80 faults 130
B.11 2KPI Synthetic data with 20/4 faults 131
B.12 DWT low-pass coefficient of 2KPI synthetic data with 20/4 faults 131
B.13 DWT high-pass coefficient of 2KPI synthetic data with 20/4 faults 132
B.14 LWT low-pass coefficient of 2KPI synthetic data with 20/4 faults 132
B.15 LWT high-pass coefficient of 2KPI synthetic data with 20/4 faults 133
B.16 3KPI Synthetic data with 20/4 faults 133
B.17 DWT low-pass coefficient of 3KPI synthetic data with 20/4 faults 134
B.18 DWT high-pass coefficient of 3KPI synthetic data with 20/4 faults 134
B.19 LWT low-pass coefficient of 3KPI synthetic data with 20/4 faults 135
B.20 LWT high-pass coefficient of 3KPI synthetic data with 20/4 faults 135
B.21 2KPI Synthetic data with 80/1 fault 136
B.22 DWT low-pass coefficient of 2KPI synthetic data with 80/1 faults 136

LIST OF FIGURES (Continued)

Figure **Page**

B.23 DWT high-pass coefficient of 2KPI synthetic data with 80/1 faults 137
B.24 LWT low-pass coefficient of 2KPI synthetic data with 80/1 faults 137
B.25 LWT high-pass coefficient of 2KPI synthetic data with 80/1 faults 138
B.26 3KPI Synthetic data with 80/1 faults 138
B.27 DWT low-pass coefficient of 3KPI synthetic data with 80/1 faults 139
B.28 DWT high-pass coefficient of 3KPI synthetic data with 80/1 faults 139
B.29 LWT low-pass coefficient of 3KPI synthetic data with 80/1 faults 140
B.30 LWT high-pass coefficient of 3KPI synthetic data with 80/1 faults 140
B.31 Histogram of INTEL dataset (temperature reading) 141
B.32 Histogram of INTEL dataset (humidity reading) 141
B.33 Histogram of INTEL dataset (voltage reading) 142
B.34 INTEL dataset (temperature reading) 142
B.35 DWT low-pass coefficient of INTEL dataset (temperature reading) 143
B.36 DWT high-pass coefficient of INTEL dataset (temperature reading) 143
B.37 LWT low-pass coefficient of INTEL dataset (temperature reading) 144
B.38 LWT high-pass coefficient of INTEL dataset (temperature reading) 144
B.39 INTEL dataset (humidity reading) 145
B.40 DWT low-pass coefficient of INTEL dataset (humidity reading) 145
B.41 DWT high-pass coefficient of INTEL dataset (humidity reading) 146
B.42 LWT low-pass coefficient of INTEL dataset (humidity reading) 146

LIST OF FIGURES (Continued)

Figure **Page**

B.43 LWT high-pass coefficient of INTEL dataset (humidity reading) 147
B.44 INTEL dataset (voltage reading) 147
B.45 DWT low-pass coefficient of INTEL dataset (voltage reading) 148
B.46 DWT high-pass coefficient of INTEL dataset (voltage reading) 148
B.47 LWT low-pass coefficient of INTEL dataset (voltage reading) 149
B.48 LWT high-pass coefficient of INTEL dataset (voltage reading) 149
B.49 Histogram of pdg2008 dataset (surface temperature reading) 150
B.50 pdg2008 dataset (ambient temperature reading) 150
B.51 Histogram of pdg2008 dataset (solar radiation reading) 151
B.52 pdg2008 dataset (surface temperature reading) 151
B.53 DWT low-pass coefficient of pdg2008 dataset (surface temperature reading) 152
B.54 DWT high-pass coefficient of pdg2008 dataset (surface temperature reading) 152
B.55 LWT low-pass coefficient of pdg2008 dataset (surface temperature reading) 153
B.56 LWT high-pass coefficient of pdg2008 dataset (surface temperature reading) 153
B.57 pdg2008 dataset (ambient temperature reading) 154

LIST OF FIGURES (Continued)

Figure **Page**

B.58 DWT low-pass coefficient of pdg2008 dataset (ambient temperature reading) 154
B.59 DWT high-pass coefficient of pdg2008 dataset (ambient temperature reading) 155
B.60 LWT low-pass coefficient of pdg2008 dataset (ambient temperature reading) 155
B.61 LWT high-pass coefficient of pdg2008 dataset (ambient temperature reading) 156
B.62 pdg2008 dataset (solar radiation reading) 156
B.63 DWT low-pass coefficient of pdg2008 dataset (solar radiation reading) 157
B.64 DWT high-pass coefficient of pdg2008 dataset (solar radiation reading) 157
B.65 LWT low-pass coefficient of pdg2008 dataset (solar radiation reading) 158
B.66 LWT high-pass coefficient of pdg2008 dataset (solar radiation reading) 158
B.67 Histogram of NAMOS dataset (fluorimeters reading) 159
B.68 Histogram of NAMOS dataset (temperature reading from sensor 1) 159
B.69 Histogram of NAMOS dataset (temperature reading from sensor 2) 160
B.70 NAMOS dataset (fluorimeters reading) 160
B.71 DWT low-pass coefficient of NAMOS dataset (fluorimeters reading) 161
B.72 DWT high-pass coefficient of NAMOS dataset (fluorimeters reading) 161
B.73 LWT low-pass coefficient of NAMOS dataset (fluorimeters reading) 162

LIST OF FIGURES (Continued)

Figure **Page**

B.74 LWT high-pass coefficient of NAMOS dataset (fluorimeters reading) 162

B.75 NAMOS dataset (temperature reading from sensor 1) 163

B.76 DWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 1) 163

B.77 DWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 1) 164

B.78 LWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 1) 164

B.79 LWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 1) 165

B.80 NAMOS dataset (temperature reading from sensor 2) 165

B.81 DWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 2) 166

B.82 DWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 2) 166

B.83 LWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 2) 167

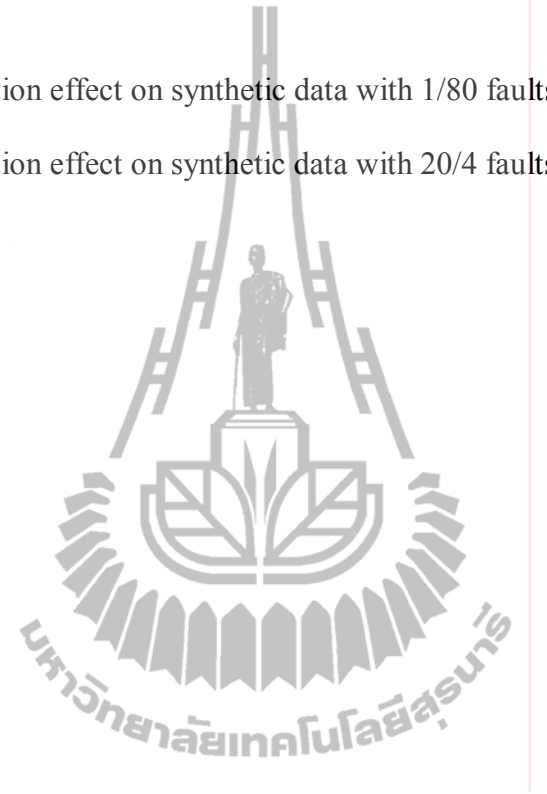
B.84 LWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 2) 167

LIST OF FIGURES (Continued)

Figure **Page**

C.1 Normalization effect on synthetic data with 1/80 faults 170

C.2 Normalization effect on synthetic data with 20/4 faults 170



SYMBOLS AND ABBREVIATIONS

WSNs	=	Wireless sensor networks
DWT	=	Discrete Wavelet Transform
LWT	=	Lifting Wavelet Transform
PCA	=	Principal Component Analysis
SOM	=	Self-Organizing Map
OCSVM	=	One-Class Support Vector Machines
KPIs	=	Key performance indicators
μ, ϖ	=	Observation index
n	=	The number of data vectors in the dataset
p, q	=	The number of parameter types or key performance indices (KPIs)
X	=	Input dataset
x^{μ}, x^{ϖ}	=	Row vector of input dataset
x	=	Sample vector from a fault-free region of the input dataset X.
x^{new}	=	A new state vector
BMU	=	Best matching unit
i	=	Neuron
t	=	Iteration index
m_i	=	Weight vector of neuron i
m_c	=	Weight vector of best matching unit
$\|\cdot\|$	=	Euclidian distance

SYMBOLS AND ABBREVIATIONS (Continued)

η_t	=	Learning rate
$h_c(i,t)$	=	Neighborhood function
$r_i(t)$	=	Positions of neurons i
$r_c(t)$	=	Positions of the BMU, c
σ	=	Radius of the neighborhood function
$\phi(\cdot)$	=	Non-linear function
X_ϕ	=	Image vectors
$\phi(x^\mu)$	=	Row vector of image vectors
R	=	Radius of sphere
ξ_μ	=	Slack variables
ν	=	Regularization parameter
$k(x^\mu, x^{\varpi})$	=	Kernel function
α_μ	=	Lagrangian multiplier
K	=	Kernel matrix
K_c	=	Center kernel matrix
1_n	=	$n \times n$ matrix of 1/n value
$\overline{X}$	=	Mean of the input dataset
PCs	=	The Principal Components
j	=	Level of wavelet transform
h_0	=	Wavelet function

SYMBOLS AND ABBREVIATIONS (Continued)

g_0	=	Scaling function
m	=	Time scaling index
f	=	Frequency translation index for wavelet level j.
L	=	Wavelet function length or Scaling function length
a_j^{DWT}	=	Current rough-scale (or approximation) DWT coefficients
a_{j+1}^{DWT}	=	Next level rough-scale (or approximation) DWT coefficients.
d_{j+1}^{DWT}	=	Next level fine-scale or detail DWT coefficients.
a_j^{LWT}	=	Current rough-scale (or approximation) LWT coefficients
a_{j+1}^{LWT}	=	Next level rough-scale (or approximation) LWT coefficients.
d_{j+1}^{LWT}	=	Next level fine-scale or detail LWT coefficients.
a	=	The amount of faults per series
s	=	The amount of series of faults
FPR	=	False positive rate
DR	=	Detection rate
LP	=	Low pass coefficients
HP	=	High pass coefficients

CHAPTER I

INTRODUCTION

This chapter introduces a background on data compression and anomaly detection in wireless sensor networks (WSNs). It also presents the motivation for applying reinforcement learning to achieve the best mutual policy for the agents which is the main focus of this thesis.

1.1 Background problem and significance of study

Nowadays, many wireless communication applications are applied in agriculture. One example is using wireless sensor nodes for agriculture monitoring thereby obtaining data measurements from wireless sensor networks (WSNs) that consist of wireless sensor nodes located at different places on a farm. Such measurements are collected and forwarded to a central server. WSNs are formed using many sensor nodes that are small and inexpensive, with an onboard simple central processing unit (CPU), limited memory, and limited energy resource. Therefore, each sensor node has many limitations such as memory, bandwidth, low-rate radio communication, energy consumption, and computational capabilities (Goh, Sim, and Ewe, 2007). These limitations make communication unreliable which can contribute to occurrences of the anomalies in a set of sensor data measurements.

An anomaly or outlier in a set of sensor data measurements is defined as an observation that appears to be inconsistent with the remainder of the dataset (Rajasegarar, Leckie, and Palaniswami, 2008). Anomalies, which occur from unusual

phenomena in monitor domain, can damage agricultural produce. Some applications, such as in a hydroponics farm that requires accurate pH level control of solution plant, or in a bio-organic fertilizer plant that requires temperature control in the fertilizer compost process, or in aquaculture monitoring that requires monitoring of the dissolved oxygen value (DO) in water, immediate anomaly detection in a set of data measurement is essential in order to take immediate course of actions.

However, due to hardware limitations WSNs require minimal energy consumption. Since radio communication in WSN consume more energy than processing and computing (Rajasegarar et al., 2008), computation with small datasets will consume smaller energy than huge datasets. Furthermore, some researches such Siripanadorn, Hattagam, and Teaumroong, (2010a, 2010b) and Kiziloren and Germen (2009) used data compression by various algorithms. And found that such approach can increase the efficiency of anomaly detection. Motivated by their findings, we will extend their algorithm to our algorithm by integrate data compression with another anomaly detection technique. The underlying aim of this research is to determine an efficient combination of data compression and anomaly detection techniques suitable for resource-constrained conditions is WSNs. The research initially focuses on the effect of data compression on anomaly detection and then proceeds to find the most efficient integrated anomaly detection and data compression techniques. A prototype of the selected integrated anomaly detection and data compression will then be developed on a WSN node for agricultural monitoring.

1.1.1 Anomaly detection techniques

In general, anomaly or outlier detection mechanisms can be categorized into three general approaches depending on the type of background

knowledge of the data available. The first approach finds outliers without prior knowledge of the underlying data. Such approach includes the family of parametric statistical anomaly detection techniques. The second approach uses supervised classification, where the classifier is trained with labeled data. The third approach is analogous to semi-supervised recognition (Rajasegarar, Leckie, and Palaniswami, 2008). The second and third approaches are referred to as nonparametric anomaly detection techniques.

1.1.1.1 The parametric statistical anomaly detection techniques

The parametric statistical anomaly detection techniques assume that the normal data is generated by a parametric distribution (Rajasegarar et al., 2008). Therefore, the density distribution of the data is known a priori. The parametric distributions such as mean, variance, probability density, are first estimated, then anomalies are flagged as those data points with low likelihood given that distribution. For example, the Chi-Square Test Statistical Method was used to detect sinkhole attacks in WSNs that use sample means as thresholds (Rajasegarar et al., 2008). The Gaussian Model Based Method was tested with a Gaussian distribution dataset by using an estimated mean for the anomaly score (Chandola, Banerjee, and Kumar, 2009). The Linear Least – Squares Estimation Method (LLSE) computed the mean and variance of sensor measurements, as well as the covariance between sensor measurements based on the training dataset, and used them to detect anomalies in the test dataset (Sharma, Golubchik, and Govindan, 2010).

However, parametric statistical techniques are suitable when the underlying type of distribution of the data is well known. This is therefore, highly application dependent. The technique becomes much harder when the sensors become

mobile rather than static, and also when the data distribution evolves over the lifetime of the network. Since WSNs have limited resources, can be moveable and their data distribution can change frequently, parametric statistical anomaly detection techniques are unsuitable for WSNs (Rajasegarar et al., 2008; Chandola et al., 2009).

1.1.1.2 Nonparametric anomaly detection techniques

Nonparametric anomaly detection techniques do not assume any prior knowledge about the distribution of the data. Therefore, these techniques are suitable for resource-constrained WSNs where the data distribution may change frequently and device can become moveable (Rajasegarar et al., 2008; Chandola et al., 2009). There are many solutions of nonparametric anomaly detection techniques in the literature. For example, Rule Based Methods like the Histogram Method was the simplest non-parametric technique that divided and plotted the time series of sensor readings into some groups of N samples to find thresholds for anomaly detection. The efficiency of such method depended on N obtained from the training phase (Sharma et al., 2010). It can quickly detect anomaly in the testing phase, and obtain a threshold in the training phase (Chandola et al., 2009). However, the open issues were to what extent anomaly conditions can be predefined, as well as how to update the rules automatically in response to changes in a dynamic environment (Rajasegarar et al., 2008).

Density Based Methods were used for distributed anomaly detection. An interesting open issue is to identify a limit for the number of children nodes for each parent node, based on the effect of the computational and communication load at the parents on the lifetime of the network (Rajasegarar et al., 2008).

Data Clustering Based Methods, such as the K-Nearest Neighbor (KNN), was proposed in (Chandola et al., 2009; Kiziloren and Germen, 2009; Rajasegarar, Leckie, Palaniswami, and Bezdek, 2006; Yao, Sharma, Golubchik, and Govindan, 2010). This method grouped a similar data into clusters of fixed-width and finds the average inter-cluster distance between clusters. The performance of this method depended strongly on the cluster width. However, data clustering methods can only identify which data vectors contain anomalies, but cannot identify anomalous readings within a data vector. The efficiency of this method performed poorly in detecting long term anomalies since this method was designed to detect outliers and did not exploit temporal correlations within the time series (Yao et al., 2010).

Cumulative Summation (CUSUM) detected changes in mean, variance, or covariance in sensor measurements without assuming any knowledge about the underlying data distribution (Yao et al., 2010). This method was computationally simple but had long training phase because this method considered each feature in isolation (Rajasegarar et al., 2008).

On the other hand, the Time Series Analysis Based Methods can be performed online. Example for such method included the Autoregressive Integrated Moving Average (ARIMA) that was used in (Chandola et al., 2009; Sharma et al., 2010; Yao et al., 2010) as a standard tool for modeling and forecasting time series data with periodicity. Temporal correlations in sensor measurements were used to construct an ARIMA model of sensor data. In order to predict the value of future readings, with new sensor readings and compared it against their predicted value. If the difference between these values was above a threshold, the new data is marked as anomalous. However, ARIMA performed poorly at detecting long duration

anomalies. An alternative method was the Segment Sequence Analysis (SSA), which was suitable for periodicity and leveraged temporal correlations in sensor measurements. SSA can detect most long duration anomalies. Furthermore, by combining SSA and Rule Based Methods together, both short and long duration anomalies can be detected. SSA was robust to the presence of sensor data faults in the reference time series, and had low computation and memory cost, therefore it can be effectively implemented in WSN. A comparison of Yao et al. (2010) showed that SSA had more efficiency than KNN, CUSUM, PCA and ARIMA. However, SSA itself alone can detect only long duration anomalies (Yao et al., 2010).

Kernel-based methods like the Support Vector Machine (SVM) were proposed for data classification and anomaly detection. SVM is a popular and useful technique for data classification (Hsu, Chang, and C. J. Lin, 2003). It has been applied for hyperspectral remote sensing image classification with good performance in recent years (Du, Tan, and Xing, 2010). The first key concept of this method was developed for binary classification problems by mean of mapping the original data vectors from input space into higher dimensional space called feature space using the kernel function (Lutsa, et al., 2010). Therefore, the kernel functions play an important role in SVM. There have been many kernel functions deployed, such as linear, polynomial, Gaussian or Radial Basis Function (RBF), and Sigmoid-shaped functions (Du et al., 2010). Therefore, SVM can be applied for anomaly detection by classifying data into normal and anomalous classes. The SVM that has been applied for anomaly detection was the One-Class Support Vector Machine (OCSVM). OCSVM was used for detecting anomalous connections in (F. Wang, Qian, Dai, and Z. Wang, 2010). Furthermore, (Laskov, Schafer, and Kotenko, 2004;

Rajasegarar et al., 2007; 2010; Y. Zhang, Meratnia, and Havinga, 2009) successfully used OCSVM to detect anomalies in WSN, with real world datasets based on fitting normal data to a quarter of sphere feature space that can change in dynamic environment. In 2008, Rajasegarar et al., suggested that OCSVM incurred little communication overhead and was suitable for sensor networks deployed in homogenous environments where the data distribution at each node was identical but unknown, and was suitable for online application. It was also noted that OCSVM had more flexibility to dynamically estimate the normal behavior from the observed feature. However, despite its performance, the approach was more demanding in terms of computational complexity than the Rule Based Methods and CUSUM. The OCSVM requires correct estimation of two parameters:

1) The kernel parameter function that maps the data to the feature space, e.g. degree in polynomial kernel or sigma (σ) in the Radial Basis Function.

2) A regularization parameter (ν), which controls the fraction of data vectors that fall inside the hyperplan or hypersphere.

The final non-parametric anomaly detection technique is the family of Learning Based Methods. Two approaches have been proposed. The first was the Hidden Markov Model (HMM) that was used to construct a model for the measurements reported by sensors in a WSN. This method was used in fault detection in which each sample was labeled either as fault-free or faulty with a particular fault type. Such labeled data was then used for estimating the parameters of the HMM (Sharma et al., 2010). However, an alternative approach which obtained detection results better than HMM was the Self-Organizing Map (SOM) (Min and Dongliang,

2009). SOM is an unsupervised neural network model for analyzing and visualizing high dimensional data into two-dimensional lattices (F. Wang, Qian, Dai, and Z. Wang, 2010). SOM was used in conjunction with OCSVM, where OCSVM was applied for anomaly detection and SOM for filtering known intrusions and classifying the unknown intrusions (F. Wang et al., 2010). Finally, they found that their model performed well by obtaining high detection rates and low false alarm rates. In 2007, Doshi, King, and Lawrence used SOM to classify hyperspectral data that was compressed by Discrete Wavelet Transform (DWT). In 2010, Xu and Chow used SOM to classify data that had been compressed with density based data reduction. In 2009, Min and Dongliang used SOM in real-time intrusion detection and founded that SOM obtained detection results better than HMM. In 2010, Siripanadorn et al. used SOM to detect anomalies in a centralized anomaly detection operation on sensor data measurements that was compressed by DWT. The authors noted that though SOM required limited storage and computing costs and can accurately detect anomalies, the processing time will increase with the size of input data. In 2009, Kiziloren and Germen used SOM to detect anomalies in network traffic once the data had been compressed with Principle Component Analysis (PCA).

Since Siripanadorn et al., (2010a; 2010b) and Kiziloren and Germen (2009) used SOM to detect anomalies in compressed data and obtain better efficiency, we are therefore motivated to compare efficiency between algorithm in (Siripanadorn et al., 2010a; 2010b; Kiziloren and Germen, 2009) and study alternative data compression techniques in WSN in order to find a suitable and efficient combination data compression and anomaly detection techniques.

Table 1.1 Ability of Anomaly Detection in Previous Works

	Previous anomaly detection works	Ability to detect faults			Adaptive to dynamic	Used with data compression
		Short	Noise	Constant		
	Parametric Statistical Anomaly Detection Technique					
1	Chi-Square Test Statistic Method	-	-	-	×	×
2	Gaussian Model Based Method	-	-	-	×	×
3	Linear Least – Squares Estimation (LLSE)	✓	✓	×	×	×
	Nonparametric Anomaly Detection Techniques					
4	Rule-Based method • Histogram Method	✓	✓	✓	×	×
5	Density Based Method	×	✓	✓	✓	×
6	Data Clustering Based Method • K-Nearest Neighbor (KNN)	✓	×	×	✓	×
7	Cumulative Summation (CUSUM)	×	✓	×	✓	×
8	Time Series Analysis Based Method • Autoregressive Integrated Moving Average (ARIMA) • Segment Sequence Analysis (SSA)	✓ ×	× ✓	× ✓	✓ ✓	× ×
9	Kernel-based method • Support Vector Machine (SVM) • One-Class Support Vector Machine (OCSVM)	- ✓	- ✓	- ×	× ✓	× ×
10	Learning Based Method • Hidden Markov Model (HMM) • Self-Organizing Map (SOM)	✓ ✓	✓ ×	× ✓	✓ ✓	× ✓

1.1.2 Data compression techniques

Many data compression techniques in WSN have been proposed in the literature. These techniques were used to reduce size of data in communication and reduce dimensional of data in order to prepare the data for suitable feature extraction for anomaly detection.

The earliest research was on performance comparison of four compression algorithms, i.e., Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT) and Vector Quantization (VQ) (Watson, Liakopoulos, Brzakovic, and Georgakis, 1995). The authors found that VQ had a better performance than other techniques and can be successfully applied online. VQ was later improved to Adaptive Learning Vector Quantization (ALVQ) in order to compress a codebook update (S. Lin, Gunopulos, Kalogeraki, and Lonardi, 2005). In addition S. Lin et al. (2005) found that ALVQ was suitable for dynamic bandwidth application. However, VQ and ALVQ were more popular used to reduce the size and detail of picture than used to reduce the dimensional of data.

More recently, a lossless compression method for WSNs was called the Sensor Lempel Ziv Welch (S-LZW) was studied and improved. Marcelloni and Vecchio (2008) improved the S-LZW to achieve a better compression ratio and lower computational complexity than the original S-LZW. Furthermore, Capo-Chichi, Guyennet, and Friedt (2009) proposed the K-Run Length Encoding (K-RLE) that offered a better compression ratio than the S-LZW. However, K-RLE used more energy than S-LZW.

Kimura and Latifi (2005) proposed three data compression algorithms in WSN, i.e., coding by ordering, in-network compression and distributed

compression. The coding by ordering method had a good compression ratio and was simple to design. However, it required a mapping table. The size of mapping table increased exponentially depending on the number of sensor nodes, therefore, unsuitable for memory-constrained sensor nodes. In 2003, Arici, Gedik, Altunbasak, and Liu first proposed the in-network compression method. After that, Boukerche and Samarah (2009) applied the in-network compression method to single-valued sensor readings. It was shown that this method did not require temporal redundancy of data for good performance. In 2008, Cai and M. Zhang used Minimum Nodes Data Gathering Tree (MNDGT) to remove some data redundancy before compression and compared the performance with the Distributed Compression. Results have shown better performance in terms of energy consumption. Since each compressed data packet did not contain the measured value, in-network compressed data may not be suitable for anomaly detection.

Sharma, Golubchik, and Govindan (2010) proposed Principle Component Analysis (PCA) which was shown to be a suitable tool for reducing the dimensionality of a dataset. It is a classical statistical method which was used to transform attributes of a dataset into a new set of uncorrelated attributes called principal components (PCs).

In addition, the Wavelet Transform (WT) has been commonly used to compress data using many types of transforms. The first method is the Data Aggregation based on Wavelet Entropy (DAWE) which used cluster members and cluster heads to reduce transmitting packets in WSN. DAWE offered better energy efficiency than clustering algorithm like Low-Energy Adaptive Clustering Hierarchy (LEACH). Since this work was based on synthetic random data which may exhibit

data redundancy different from real sensor measurements, energy efficiency may not be guaranteed for in real sensor measurements (Bruce, Koger, and J. Li, 2002). The second wavelet transform is the Discrete Wavelet Transform (DWT). Many researchers still used DWT to compress data although S. Lin et al. (2005) showed that VQ gave more efficiency. Since DWT can reduce the dimension of data while still preserving significant features of the data, it can be used with other application such as anomaly detection, whereas VQ can only reduce size and detail of data (S. Lin et al. 2005). DWT was used to reduce the dimension of hyperspectral data and outperformed techniques that considered only the frequency content of the signal but not localized information like discrete cosine transform (DCT) (X. L. Li, J. W. Zhang, and W. H. FANG, 2009). DWT was used for preprocessing data to reduce the dimension of hyperspectral data prior to feeding it into SOM for classification purposes (Xu and Chow, 2010). DWT was used to compress data before feeding the data to SOM for anomaly detection in WSNs in (Siripanadorn, et al., 2010a; 2010b). The integrated SOM and DWT outperformed the SOM in term of anomaly detection performance. Finally, the Lifting Wavelet Transform (LWT) has been found to perform well in terms of energy savings, when compared its performance with the data compression algorithm based on traditional wavelet transforms like DWT. Furthermore, LWT when used as a distributed wavelet compression algorithm, the authors found it performed constantly well in (Ciancio and Ortega, 2005; Manjunath and Ravikumar, 2010). Therefore, LWT warrants potential use in detection.

To the best of our knowledge, SOM has been an anomaly detection method that has shown good performance. Furthermore, there have been algorithms which used SOM for anomaly detection in conjunction with data compression

schemes like DWT and PCA. These integrated algorithms have been found to increase the efficiency of anomaly detection. On the other hand, anomaly detection using OCSVM has been considered easier to implement than SOM (Du, Tan, and Xing, 2010). It has also been a popular and useful technique for anomaly detection due to its flexibility and small energy consumption. OCSVM are deemed suitable for sensor networks and online application. Furthermore, the Lifting Wavelet Transform (LWT) was shown to provide a better data compression technique than DWT.

Therefore, this research is focused on incorporating the wavelet-based data compression scheme with OCSVM anomaly detection (OCSVM + DWT and OCSVM + LWT). Previous works of Siripanadorn, et al., (2010a; 2010b) motivated us to compare efficiency with their algorithm (SOM + DWT). Furthermore, the proposed OCSVM + LWT algorithm was also compared with other variants of data compression and anomaly detection schemes, including the OCSVM alone (uncompressed data), the OCSVM + DWT, and the OCSVM + PCA algorithm in order to find the most efficient algorithm for WSNs.

1.2 Research objectives

1.2.1 To study efficiency in anomaly detection and data compression in wireless sensor networks.

1.2.2 To learn effects of data compression on anomaly detection.

1.2.3 To find a combination of data compression and anomaly detection scheme most suitable for implementation in a wireless sensor node.

1.3 Research hypothesis

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

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

1.3.3 Data compression affects the ability to detect anomalies in data.

1.4 Basic agreements

1.4.1 MATLAB is used to generate synthetic data and simulate the anomaly detection algorithm in a WSN.

1.4.2 Anomaly detection algorithm will detect real-world faults which are categorized into three types as follows;

1) Short fault; a sharp change in the measurement value between two successive data points and affect a single sample at a time,
2) Noisy fault; a fault that occurs when variance of the sensor readings increases and affects a number of successive samples
3) Constant fault; a fault that occurs when reports a constant value for a large number of successive samples.

1.5 Scope and limitation

1.5.1 Anomaly detection methods for WSNs were studied.

1.5.2 Data compression methods for WSNs were studied.

1.5.3 Anomaly detection and data compression were studied to detect abnormal data in both synthetic data and real world data.

1.5.4 Results from simulation of combined anomaly detection and data compression algorithm were analyzed and concluded.

1.6 Expected benefit

1.6.1 To obtain an efficient integrated anomaly detection and data compression algorithm for WSNs.

1.7 Synopsis of Thesis

The remainder of this thesis is organized as follows.

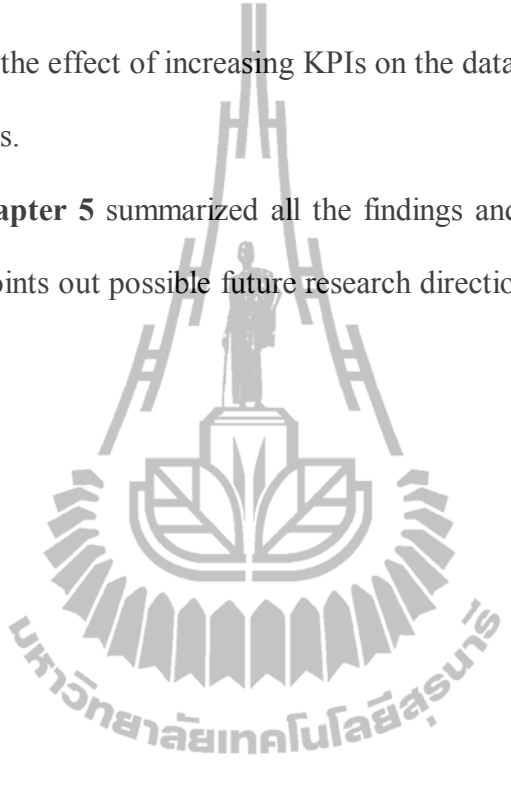
Chapter 2 presented the theoretical background which is the foundation of the contributions of this thesis. Firstly, the anomaly detection techniques in related works were presented. This was followed by the self-organizing map (SOM) algorithm and the one-class support vector machines (OCSVM). Finally, the data compression techniques in related works were presented, followed by the principal components analysis (PCA), discrete wavelet transforms (DWT) and lifting wavelet transforms (LWT).

Chapter 3 presented the experiments conducted to evaluate the performance of our first proposed algorithm (OCSVM + DWT) and compared with the previous algorithm (SOM + DWT). The experiments evaluated the anomaly detection and data compression methods described in chapter 2 with series of synthetic data injected by various synthetic faults. Furthermore, performance evaluation with the real-world datasets with real faults from various sensor networks was also presented.

Chapter 4 presented the experiments conducted to evaluate the performance of our second proposed algorithm (OCSVM + LWT) and compared with other

variants of integration such as OCSVM + DWT and OCSVM + PCA. The experiments evaluated their performance with series of synthetic data and real-world datasets as in chapter 3. In addition, we extended the experiments to both synthetic data and real-world datasets which consisted of more KPIs than the dataset in chapter 3, in order to study the effect of increasing KPIs on the data compression and anomaly detection techniques.

Finally, **Chapter 5** summarized all the findings and the original contributions in this thesis and points out possible future research directions.



CHAPTER II

BACKGROUND THEORY

In the previous chapter, we described the reason for choosing the data compression and anomaly detection techniques which were used in this thesis. The data compression techniques include the discrete wavelet transforms (DWT), lifting wavelet transforms (LWT) and principal component analysis (PCA). The anomaly detection techniques include the self-organizing map (SOM) and one-class support vector machines (OCSVM). In this chapter, we presented the theoretical background which is the foundation of the contributions of this thesis. Firstly, the anomaly detection techniques in related works are presented. These are then followed by the introduction to the SOM and the OCSVM. The data compression techniques in related works are presented next. Finally, a concise introduction on the DWT, LWT and PCA is provided.

2.1 Anomaly Detection

An anomaly or outlier in a set of sensor data measurements is defined as an observation that appears to be inconsistent with the remainder of the dataset (Rajasegarar, Leckie, and Palaniswami, 2008). Anomalies, which occur from unusual phenomena in monitor domain, can damage agricultural produce. In 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 $x^{\mu} \in \Re$

$$x^{\mu} = \begin{bmatrix} x_1^{\mu} \\ x_2^{\mu} \\ x_3^{\mu} \\ \vdots \\ x_p^{\mu} \end{bmatrix} = \begin{bmatrix} KPI_1^{\mu} \\ KPI_2^{\mu} \\ KPI_3^{\mu} \\ \vdots \\ KPI_p^{\mu} \end{bmatrix} \tag{2.1}$$

where μ is the observation index: $\mu = 1,2,3,...,n$

n is the number of data vectors in the dataset,

p 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 newly state vector x^{new} as normal or abnormal.

In general, anomaly or outlier detection mechanisms can be categorized into three general approaches depending on the type of background knowledge of the data available as described in chapter I. Since the nonparametric anomaly detection techniques do not assume any prior knowledge about the distribution of the data, these techniques are suitable for resource-constrained WSNs where the data distribution may change frequently and device can become moveable (Rajasegarar, et al., 2008; Y. Zhang, Meratnia, and Havinga, 2009). In this chapter, we discuss two popular anomaly detection techniques called the kernel-based method such as OCSVM, and a learning-based method such as SOM.

2.1.1 Self-organizing map (SOM)

SOM is an unsupervised neural network model for analyzing and visualizing high dimensional data manifold into two-dimensional lattices, grid display (F. Wang, Qian, Dai, and Z. Wang, 2010). SOM can extract statistical regularities from the input data vectors and encode them in weight vectors using unsupervised learning.

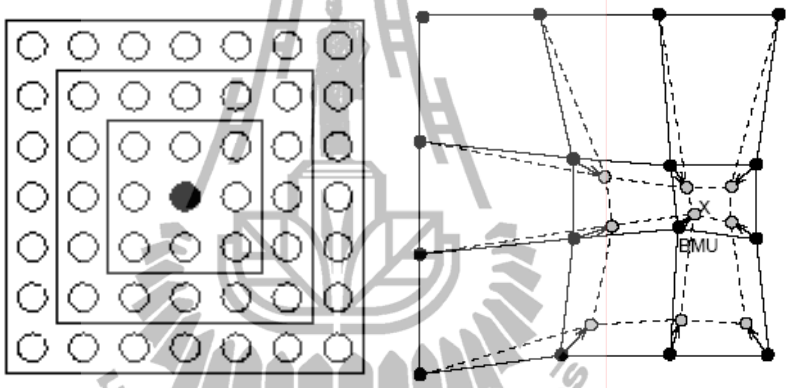


Figure 2.1 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3 neighborhoods, (b) SOM updates the BMU with 1- Neighborhood.

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).

Neuron i maintains a weight vector 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 ×) 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 x which was chosen from a fault-free region of the input dataset $X = \{x^{\mu} : \mu = 1,2,3,\ldots,n\}$.

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 $m_i(t)$ is the closest to the input vector $x(t)$. We denote such weight vector by $m_c(t)$ and refer to it as the Best-Matching Unit (BMU), which is

$$\|x(t) - m_c(t)\| = \arg\min_{\forall i} \|x(t) - m_i(t)\| \tag{2.2}$$

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

$$m_i(t+1) = m_i(t) + \eta_t h_c(i,t)[x(t) - m_i(t)] \tag{2.3}$$

where t is the iteration index.

$x(t)$ 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

$$h_c(i,t) = \exp\left(-\frac{\|r_c(t) - r_i(t)\|^2}{2\sigma^2(t)}\right) \tag{2.4}$$

where $r_i(t)$ is the positions of neurons *i*.

$r_c(t)$ is the positions of the BMU *c*.

$\sigma(t)$ 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 (Siripanadorn, Hattagam, and Teaumroong, 2010).

There are many research used SOM for many application. SOM was used in conjunction with OCSVM, where OCSVM was applied for anomaly detection and SOM for filtering known intrusions and classifying the unknown intrusions in (F. Wang, et al., 2010). Their model performed well by obtaining high detection rates and low false alarm rates. In 2007, Doshi, King, and Lawrence used SOM to classify hyperspectral data that was compressed by Discrete Wavelet Transform (DWT). In 2009, Min and Dongliang used SOM in real-time intrusion detection and founded that SOM obtained detection results better than HMM. In addition, Kiziloren and Germen

used SOM to detect anomalies in network traffic once the data had been compressed with Principle Component Analysis (PCA). In 2010, Siripanadorn, et al. used SOM to detect anomalies in a centralized anomaly detection operation on sensor data measurements that was compressed by DWT. The authors noted that though SOM required limited storage and computing costs and can accurately detect anomalies, the processing time will increase with the size of input data.

Due to work of Min and Dongliang (2009); Siripanadorn, et al. (2010) use SOM to detect anomalies in compressed data and obtain better efficiency, we are therefore motivated to compare efficiency between algorithm in their work and study alternative data compression techniques in WSN in order to find a suitable and efficient combination data compression and anomaly detection techniques.

2.1.2 One-class support vector machines (OCSVM)

In 2004, Tax and Duin applied the one-class support vector machines (OCSVM) from the Support Vector Machine (SVM) for outlier detection. The first key concept of the OCSVM is to map the original data vectors from input space into higher dimensional space called feature space using the kernel function (Lutsa, et al., 2010). Therefore, the kernel functions play an important role in both SVM and OCSVM. There have been many kernel functions deployed, such as linear, polynomial, Gaussian or Radial Basis Function (RBF), and Sigmoid-shaped functions (Du, Tan, and Xing, 2010).

Later, P. Laskov, C. Schafer and I. Kotenko (2004) have extended this approach into a special type of SVM call Quarter-Sphere OCSVM. The key idea of the Quarter-Sphere OCSVM algorithm is to encompass the data with a hypersphere anchored at the center of mass of the data in feature space. Here, we provide the

mathematical formulation of the one-class quarter-sphere SVM (Laskov, Schafer, and Kotenko, 2004).

The OCSVM requires correct estimation of two parameters. The first is the kernel parameter function that maps the data to the feature space, e.g. degree in polynomial kernel or sigma (σ) in the Radial Basis Function. The second is the regularization parameter (ν), which controls the fraction of data vectors that fall inside the hyperplane or hypersphere.

Consider an input dataset $X = \{x^{\mu} : \mu = 1,2,3,\ldots,n\}$ of p variate data vector $x^{\mu} = [x_1^{\mu}, x_2^{\mu}, x_3^{\mu}, ..., x_p^{\mu}]$ in the input space $\Re^p$ where the number of data vector in a dataset X is n. In principle, X is mapped to a feature space $\Re^q$ via a nonlinear function $\phi(\cdot)$, resulting in a set of the image vectors $X_{\phi} = \{\phi(x^{\mu}) : \mu = 1,2,3,\ldots,n\}$ where a row vector of image vectors is $\phi(x^{\mu}) = [\phi(x_1^{\mu}), \phi(x_2^{\mu}), \phi(x_3^{\mu}), ..., \phi(x_q^{\mu})]$. The aim is to fit a hypersphere in a feature space with minimum effective radius $R > 0$, centered at the origin, encompassing a majority of the image vectors X_{ϕ}. This can be formulated as an optimization problem as follows (Laskov, et al., 2004; Rajasegarar, et al., 2008):

$$\min_{R \in \Re^+, \xi \in \Re} \quad R^2 + \frac{1}{\nu n} \sum_{\mu=1}^{n} \xi_{\mu}$$

Subject to: $\left\| \phi(x^{\mu}) \right\|^2 \leq R^2 + \xi_{\mu}$ (2.5)

$$\xi_{\mu} \geq 0$$

where ξ_μ are the slack variables that allow some of the image vectors to lie outside the sphere. The parameter $\nu \in (0,1)$ is the regularization parameter which controls the fraction of image vectors that lie outside the sphere, i.e., the fraction of image vectors that can be anomalies. $\left\|\phi(x^\mu)\right\|^2$ is the inner product of the image vector $\phi(x^\mu)$ and can be replacing by a linear kernel function $k(x^\mu, x^\mu)$. Therefore, $k(x^\mu, x^\mu) = \left\|\phi(x^\mu)\right\|^2$ can be used to compute the similarity of any two vectors in the feature space using the original attribute set. Equation (2.6) is the dual formulation of the primal problem in equation (2.5) which can be obtained as follows (Laskov, et al., 2004; Rajasegarar, et al., 2008):

$$\min_{\alpha \in \Re} -\sum_{\mu=1}^{n} \alpha_\mu k(x^\mu, x^\mu)$$

$$\text{Subject to:} \quad \sum_{\mu=1}^{n} \alpha_\mu = 1 \tag{2.6}$$

$$0 \le \alpha_\mu \le \frac{1}{\nu n}$$

where $\alpha_\mu \ge 0$ is a Lagrangian multiplier.

This dual problem (2.6) is a linear optimization problem. In order to solve this problem, the image vectors in the feature space are centered in the space using center kernel matrix as follows:

$$K_c = K - 1_n K - K\, 1_n + 1_n K\, 1_n \tag{2.7}$$

where K is an $n \times n$ kernel metric consist of $k(x^{\mu}, x^{\varpi})$ where $\mu, \varpi = 1,2,3,...,n$. If $\mu = \varpi$, we can obtain $k(x^{\mu}, x^{\varpi}) = k(x^{\mu}, x^{\mu})$. Therefore, we can obtain $k(x^{\mu}, x^{\mu})$ from the norms of image vector $\phi(x^{\mu})$. Otherwise, $k(x^{\mu}, x^{\varpi})$ can be obtained from the kernel function, such as linear, polynomial, RBF kernel. Furthermore, 1_n is an $n \times n$ metric with all values equal to $1/n$. Once the image vectors are centered, the norms of the kernels are no longer equal. Hence, the dual problem (2.7) can now be solved.

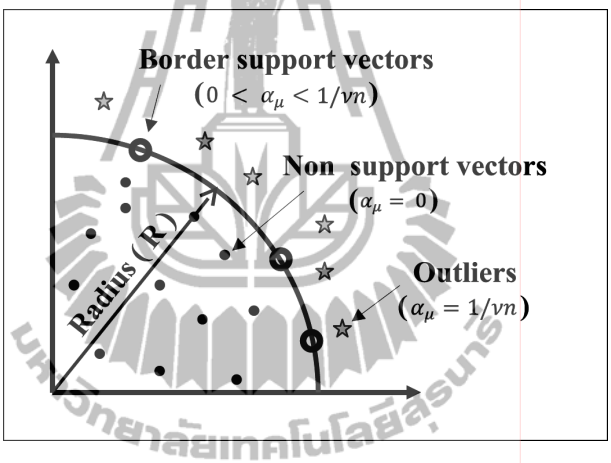


Figure 2.2 Geometry of the quarter-sphere OCSVM

The $\{\alpha_{\mu}\}$ can be obtained using widely available linear optimization techniques. The image vectors can be classified as shown in figure 2.2. The image vectors with $\alpha_{\mu} = 0$ will fall inside the sphere. The image vectors with $\alpha_{\mu} > 0$ are called the *support vectors.* Support vectors with $\alpha_{\mu} = 1/vn$ are termed *outliers*, which fall outside the sphere. Support vectors with $0 < \alpha_{\mu} < 1/vn$ reside on the surface of the sphere, and hence are called the *border support vectors*. Moreover, the radius of the sphere R can be obtained using $R^2 = k(x^{\mu}, x^{\mu})$, for any border support vector x^{μ}.

F. Wang, Qian, Dai, and Z. Wang, (2010) used OCSVM for detecting anomalous connections. Furthermore, Laskov, et al. (2004), Rajasegarar, et al. (2007; 2008; 2010) and Y. Zhang, et al. (2009) successfully used OCSVM to detect anomalies in WSN, with real world datasets based on fitting normal data to a quarter of sphere feature space that can change in dynamic environment. In 2008, Rajasegarar, et al. suggested that OCSVM incurred little communication overhead and was suitable for sensor networks deployed in homogenous environments where the data distribution at each node was identical but unknown, and was suitable for online application.

2.2 Data Compression

Since our research was focused on incorporating data compression techniques with anomaly detection techniques, in order to minimize the energy consumption in WSNs, we were interested in studying the effects of data compression on anomaly detection. Data compression techniques were used to reduce size of data prior to transmission and perform feature extraction on the dataset prior to anomaly detection,

Data compression techniques, which when integrated with anomaly detection, can increase the efficiency of anomaly detection, are principle component analysis (PCA) and discrete wavelet transforms (DWT). In addition, the lifting wavelet transform (LWT) was shown to outperform DWT (X. L. Li, J. W. Zhang, and W. H. FANG, 2009; Manjunath and Ravikumar, 2010). Therefore, in this section the PCA, DWT and LWT algorithms will be presented.

2.2.1 Principal Component Analysis (PCA)

PCA is a classical statistical technique which has found application in fields such as noise rejection, visualization, face recognition, image compression and data compression (Smith, Online, 2002). PCA is completely reversible, making it a suitable tool for reducing the dimensionality of a dataset, while still protecting as much of the dataset as possible (Kiziloren and Germen, 2009; Dwinnell, Online, 2010).

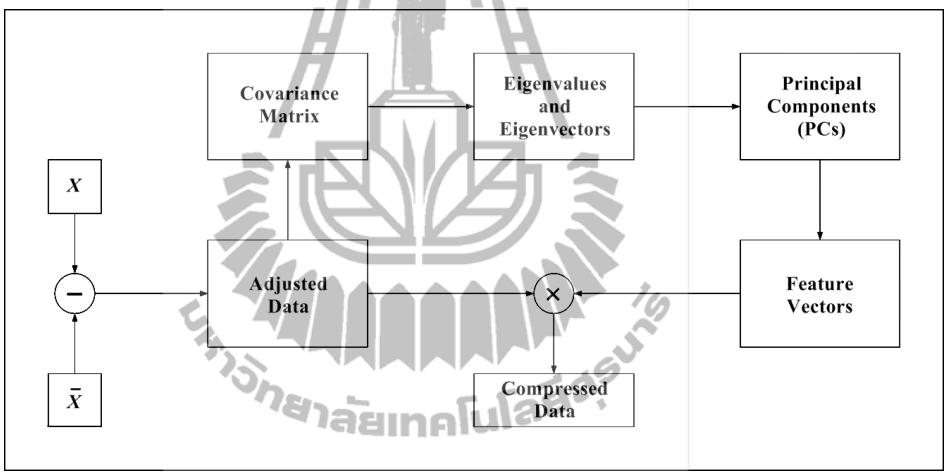


Figure 2.3 Principal component analysis (PCA) framework.

PCA was used to transform attributes of an original dataset into a new set of uncorrelated attributes called principal component (PCs) with the same number of attributes as the original dataset (Dwinnell, Online, 2010). The step to perform the PCA on the dataset is shown in Figure 2.3 and can be described as follows:

Step 1, find the mean of input dataset $X = \{x^{\mu} : \mu = 1,2,3,\ldots,n\}$ for each data dimension (KPI). Note that the original dataset X has n observation index components and p KPIs, ($n \times p$ matrix). In order to go to next step, the mean of input

dataset $\overline{X}$ which is an $n \times p$ matrix is determined.

Step 2, subtract the mean from each data dimension, resulting in the adjusted data which is a $n \times p$ matrix.

Step 3, calculate the covariance matrix of the $n \times p$ adjusted data matrix previously found in step 2. The covariance matrix is a square $p \times p$ matrix of which $\frac{p!}{2(p-2)!}$ different covariance values can be calculated (Smith, Online, 2002).

Step 4, calculate the eigenvalues and eigenvectors of the covariance matrix and order the eigenvectors by the eigenvalues from highest to lowest. The eigenvalues are in order of significance which helps in classifying the lesser significance components. The $p \times p$ covariance matrix gives p eigenvalues and $p \times p$ eigenvectors.

Step 5, select the principal components (PCs). The components of lesser significance can be ignored. The smaller the eigenvalues ignored, the less the information is lost. Therefore, if we require q KPIs data, we select the eigenvectors of the first q eigenvalues to form a feature vector matrix which is a $p \times q$ matrix.

Step 6, multiply the $n \times p$ matrix of adjusted data with the $p \times q$ matrix of feature vectors to obtain the $n \times q$ matrix of the final dataset or compressed data.

In the PCA algorithm, we found that the total elements required for the computation of PCA is $3np + 2p^2 + p + 2pq + nq$ elements. If each element is a double type therefore PCA computation uses $8 \times \left[3np + 2p^2 + p + 2pq + nq \right]$ bytes of storage space.

2.2.2 Discrete wavelet transforms (DWT)

Wavelets are mathematical functions that satisfy certain mathematical requirements and are used in representing data. Wavelets cut up data into different frequency components or resolutions or scale. The data signal can be separated into fine-scale information known as high pass (detail) coefficients, and rough-scale information known as low pass (approximate) coefficients. The original data signal can be represented in terms of wavelet coefficients. Therefore, data operations can be performed using just the corresponding wavelet coefficients.

The major advantage of DWT 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 (Siripanadorn, et al., 2010):

$$a_{j+1}^{DWT}(f) = \sum_{m} h_0(m-2f)a_j^{DWT}(f) \tag{2.8}$$

$$d_{j+1}^{DWT}(f) = \sum_{m} g_0(m-2f)a_j^{DWT}(f) \tag{2.9}$$

where a_j^{DWT} is the current rough-scale (or approximation) coefficients.

h_0 is the wavelet function.

g_0 is the scaling function.

m is the time scaling index.

f is the frequency translation index for wavelet level j.

a_{j+1}^{DWT} is the next level rough-scale (or approximation) coefficients.

d_{j+1}^{DWT} is the next level fine-scale or detail coefficients.

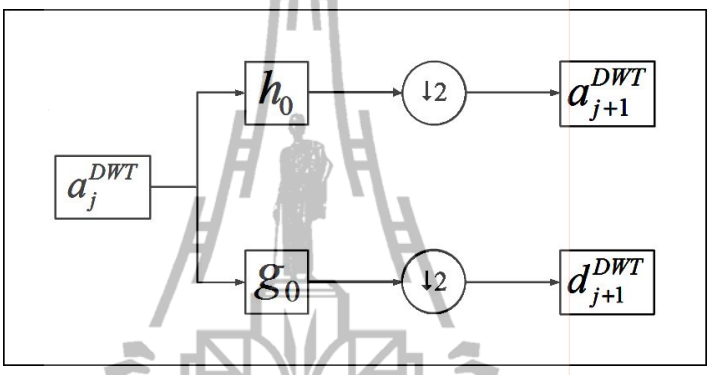


Figure 2.4 Discrete wavelet transforms (DWT) framework.

Figure 2.4 shows the DWT framework. In the first step, the rough-scale (or approximation) coefficients a_j^{DWT} which has n observation index components and p KPIs ($n \times p$ matrix), are convolved separately with h_0 and g_0, the wavelet function and scaling function each of length L, respectively, resulting in two $[(n \times L) - 1] \times p$ matrices of coefficients. After that, 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} which has $\frac{n}{2}$ observation index components and p KPIs thereby forming a $\frac{n}{2} \times p$ matrix (Siripanadorn, et al., 2010).

In the DWT algorithm, the total number of elements required for DWT computation is $2np(1+L)+2(L+p)$ elements. Assuming that each element is a double type, therefore DWT uses $8\times\left[2np(1+L)+2(L+p)\right]$ bytes of memory space

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 features of the data by replacing the data with its hierarchical coefficients, i.e., its low pass and high pass coefficients. However, DWT calculation involve convolutions, which require a large number of arithmetic computation and large storage space than a newer mathematical formulation for wavelet transform described in the next section.

2.2.3 Lifting wavelet transforms (LWT)

In 1998, Sweldens proposed a new mathematical formulation for wavelet transform called lifting wavelet transform (LWT). The main feature of LWT is to convert the filter implementation of DWT into band matrix multiplication which requires fewer computations. A comparison of computational requirements with the convolution based DWT was provided in (Achaya and Chakrabarti, 2006).

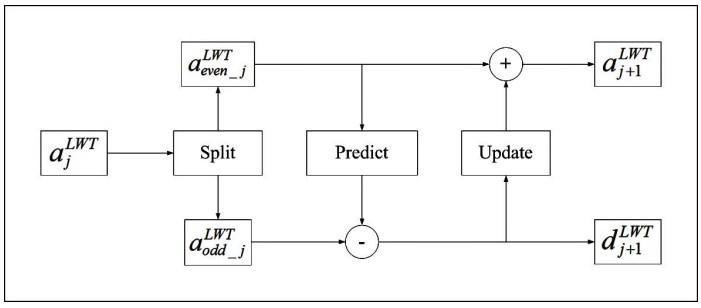


Figure 2.5 Lifting Wavelet Transform (LWT) framework.

Figure 2.5 shows the LWT framework. The LWT algorithm divides the input dataset X or current rough-scale (or approximation) coefficients a_j^{LWT} into 3 stages as follows:

1) Split: the input dataset or current rough-scale (or approximation) coefficients a_j^{LWT} which has n observation index components and p KPIs ($n \times p$ matrix), was split into an even half $a_{even\_j}^{LWT}$ and an odd half $a_{odd\_j}^{LWT}$ elements each forming a $\frac{n}{2} \times p$ matrix.

2) Predict: the odd half $a_{odd\_j}^{LWT}$ coefficients are then predicted by a linear combination of even half $a_{even\_j}^{LWT}$ coefficients, thus producing a prediction error or high pass (fine-scale or detail) coefficients d_{j+1}^{LWT} which is a $\frac{n}{2} \times p$ matrix.

3) Update: the even half coefficients $a_{even\_j}^{LWT}$ were then updated by adding them to a linear combination of the prediction error or high pass (fine-scale, detail) coefficients d_{j+1}^{LWT}, resulting in a set of low pass (approximation) coefficients a_{j+1}^{LWT} which is a $\frac{n}{2} \times p$ matrix.

LWT has an in-place computation therefore it does not require extra buffer memory (Achaya and Chakrabarti, 2006). The total number of elements required for the calculation of LWT is $4np$ elements. Assuming that each element is a double type, therefore LWT uses $8 \times 4np$ bytes of memory space, which is significantly fewer than DWT which required $8 \times \left(2np(1+L) + 2(L+p)\right)$ bytes in memory where L is the length of filter or the wavelet function and scaling function and $L > 1$.

2.3 Summary

In this chapter, we presented an overview of anomaly detection and data compression scheme in wireless sensor networks. In particular, the self-organizing map (SOM) and one-class support vector machines (OCSVM) anomaly detection techniques were highlighted. Furthermore, the principal component analysis (PCA), discrete wavelet transform (DWT) and lifting wavelet transform (LWT) data compression techniques were also described.

The self-organizing map (SOM) performed well in terms of anomaly detection. However, SOM demanded intensive storage and computing requirements which deemed costly for onboard implementation for sensor nodes. Although DWT can accurately detect anomalies, the processing time increases with the size of input data.

The one-class support vector machines (OCSVM) was used to detect anomalous connections and successfully deployed to detect anomalies in WSN online with real world datasets. Furthermore, OCSVM incurred little communication overhead and was suitable for sensor networks deployed in homogenous environments where the data distribution at each node was identical but unknown.

The principal component analysis (PCA) is suitable tool for reducing the dimensionality of a dataset while maintaining a small number of attributes of the dataset. However, PCA has complex computation.

Wavelet transform separates data into fine-scale information known as high pass (detail) coefficients, and rough-scale information known as low pass (approximate) coefficients. In this chapter, we described the discrete wavelet transform (DWT) and the lifting wavelet transform (LWT). DWT involves the convolutions while LWT

requires matrix multiplication. Consequently, DWT has a larger number of arithmetic computations and demands more storage space than LWT.

CHAPTER III

DISCRETE WAVELET TRANSFORM AND ONE-CLASS SUPPORT VECTOR MACHINES FOR ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS

Data readings from wireless sensor networks (WSNs) may be abnormal due to detection of unusual phenomena, limited battery power, 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. This section proposes an integrated data compression and anomaly detection algorithm in WSNs which can detect anomalies accurately by employing half of sensor data measurement, instead of using all the sensor data measurement. The contribution of this chapter are centered on data compression by using Discrete Wavelet Transform (DWT) then feeding to anomaly detection by using One-Class Support Vector Machine (OCSVM). We tested our algorithm with several synthetic and real world datasets. After that, we compared the efficiency of our algorithm with the previous techniques such the self-organizing map (SOM) with DWT. Finally, we found that the proposed algorithm outperformed previous techniques in terms of near 100% detection rate (DR) and with marginal increase in false positive rates (FPR) in presence of short and noise faults.

3.1 Introduction

Wireless sensor networks (WSNs) consist of wireless sensor nodes located at different places in an area of interest. Data measurements are collected by these sensor nodes and forwarded to a central server. WSNs are formed using many sensor nodes that have many limitations such as memory, bandwidth, energy consumption, and computational capabilities (Goh, Sim, and Ewe, 2007). These limitations make communication unreliable which can contribute to occurrences of the anomalies in a set of sensor data measurements.

An anomaly or outlier in a set of sensor data measurements is defined as an observation that appears to be inconsistent with the remainder of the dataset (Rajasegarar, Leckie, and Palaniswami, 2008). Anomalies, which occur from unusual phenomena in monitor domain, can damage agricultural produce. Some applications, such as in a hydroponics farm that requires accurate pH level control of solution plant, or in a bio-organic fertilizer plant that requires temperature control in the fertilizer compost process, immediate anomaly detection in a set of data measurement is essential in order to take immediate course of actions.

However, due to hardware limitations WSNs require minimal energy consumption. Since radio communication in WSN consumes more energy than processing and computing (Rajasegarar et al., 2008). Siripanadorn, Hattagam, and Teaumroong (2010a; 2010b) used data compression by Discrete Wavelet Transform (DWT) prior to feeding data to an anomaly detection algorithm. Such approach was found to increase the efficiency of anomaly detection. Motivated by their findings, we extend their study to integrate DWT data compression with an alternative anomaly detection technique.

The One-Class Support Vector Machine (OCSVM) is a popular and useful anomaly detection technique that does not assume any prior knowledge about the distribution of the data and has been found suitable for resource-constrained WSNs (Wang, Qian, Dai, and Wang, 2010). OCSVM can update the normal behavioral model of the sensed data in an online manner. Laskov, Schafer, and Kotenko (2004); Rajasegarar et al. (2007; 2008; 2010); and Zhang, Meratnia, and Havinga (2009) successfully used OCSVM to detect anomalies in WSNs, with real world datasets based on fitting normal data to a quarter of sphere feature space that can change in dynamic environment. However, to the best of our knowledge, the integration between OCSVM and DWT has not yet been proposed. Therefore, the underlying aim of this chapter is to study the effect and efficiency of DWT data compression on the OCSVM anomaly detection technique and assess its suitability for deployment in resource-constrained conditions in WSNs.

3.2 Anomaly Detection

The first step of anomaly detection involves selecting the data parameters to be monitored and grouping them together in a pattern vector $x^{\mu} \in \Re^{p}$, $\mu = 1,2,...,n$

$$x^{\mu} = \begin{bmatrix} x_1^{\mu} \\ x_2^{\mu} \\ x_3^{\mu} \\ \vdots \\ x_p^{\mu} \end{bmatrix} = \begin{bmatrix} KPI_1^{\mu} \\ KPI_2^{\mu} \\ KPI_3^{\mu} \\ \vdots \\ KPI_p^{\mu} \end{bmatrix} \tag{3.1}$$

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

3.2.1 One-Class Support Vector Machines (OCSVM)

In 2004, Tax and Duin have proposed a one-class support vector machines (OCSVM) formulation for outlier detection. Then Laskov, Schafer, and Kotenko (2004) have extended this approach into a special type of SVM call "Quarter-Sphere OCSVM". The key idea of this algorithm is to encompass the data with a hypersphere anchored at the center of mass of the data in feature space. Here we provide the mathematical formulation of the one-class quarter-sphere SVM.

Consider an input dataset $X = \{x^{\mu} : \mu = 1,2,3,...,n\}$ of p variate data vector $x^{\mu} = [x_1^{\mu}, x_2^{\mu}, x_3^{\mu}, ..., x_p^{\mu}]$ in the input space $\Re^p$ where the number of data vector in a dataset X is n. In principle, X is mapped to a feature space $\Re^q$ via a nonlinear function $\phi(\cdot)$, resulting in a set of the image vectors $X_{\phi} = \{\phi(x^{\mu}) : \mu = 1,2,3,\ldots,n\}$ where a row vector of image vectors is $\phi(x^{\mu}) = [\phi(x_1^{\mu}), \phi(x_2^{\mu}), \phi(x_3^{\mu}), ..., \phi(x_q^{\mu})]$. The aim is to fit a hypersphere in a feature space with minimum effective radius $R > 0$, centered at the origin, encompassing a majority of the image vectors X_{ϕ}. This can be formulated as an optimization problem as follows (Laskov et al., 2004):

$$\min_{R \in \Re^+, \xi \in \Re} \quad R^2 + \frac{1}{\nu n} \sum_{\mu=1}^{n} \xi_{\mu}$$

Subject to: $k(x^{\mu}, x^{\mu}) \le R^2 + \xi_{\mu}$ (3.2)

$$\xi_{\mu} \ge 0$$

where ξ_{μ} are the slack variables that allow some of the image vectors to lie outside

the sphere. The parameter $\nu \in (0,1)$ is the regularization parameter which controls the fraction of image vectors that lie outside the sphere, i.e., the fraction of image vectors that can be anomalies. Note that $k(x^{\mu}, x^{\mu}) = \phi(x^{\mu}) \cdot \phi(x^{\mu})^{T}$ for a Mercer kernel and $k(x^{\mu}, x^{\mu})$ is a kernel function which was used to compute the similarity of any two vectors in the feature space using the original attribute set. Equation 3.3 is the dual formulation of the primal problem in equation 3.2 which can be obtained as follows (Laskov et al., 2004):

$$\min_{\alpha \in \Re} -\sum_{\mu=1}^{n} \alpha_{\mu} k(x^{\mu}, x^{\mu})$$

Subject to: $\sum_{\mu=1}^{n} \alpha_{\mu} = 1$ (3.3)

$$0 \le \alpha_{\mu} \le \frac{1}{\nu n}$$

where $\alpha_{\mu} \ge 0$ is a Lagrangian multiplier. $\mu = 1,2,...,n$. This dual problem (3.3) is a linear optimization problem. In order to alleviate this problem, the image vectors in the feature space are centered in the space using center kernel matrix as follows (Laskov et al., 2004):

$$K_c = K - 1_n K - K\, 1_n + 1_n K\, 1_n \tag{3.4}$$

where K is an $n \times n$ kernel metric consist of $k(x^{\mu}, x^{\varpi})$ where $\mu, \varpi = 1,2,3,...,n$. If $\mu = \varpi$, we can obtain $k(x^{\mu}, x^{\varpi}) = k(x^{\mu}, x^{\mu})$. Therefore, we can obtain $k(x^{\mu}, x^{\varpi})$ from

the norms of image vector $\phi(x^{\mu})$. Otherwise, $k(x^{\mu}, x^{\varpi})$ can obtain from the kernel function, such as linear, polynomial, RBF kernel. Furthermore, 1_n is an $n \times n$ metric with all values equal to $1/n$. Once the image vectors are centered, the norms of the kernels are no longer equal. Hence the dual problem (2.7) can now be solved.

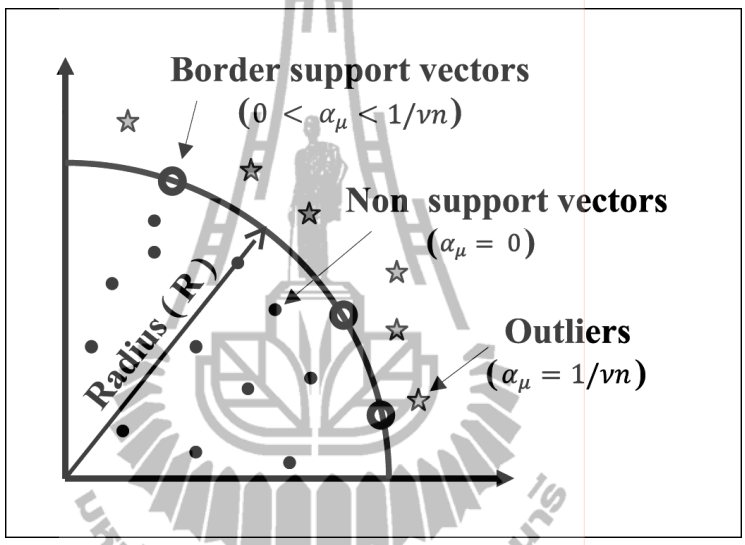


Figure 3.1 Geometry of the quarter-sphere OCSVM

The $\{\alpha_\mu\}$ can be obtained using widely available linear optimization techniques. The image vectors can be classified as Figure 2.2. The image vectors with $\alpha_\mu = 0$ will fall inside the sphere. The image vectors with $\alpha_\mu > 0$ are called the *support vectors*. Support vectors with $\alpha_\mu = 1/vn$ are termed as *outliers*, which fall outside the sphere. Support vectors with $0 < \alpha_\mu < 1/vn$ will reside *on* the surface of the sphere, and hence are called the *border support vectors*. Moreover, the radius of the sphere R can be obtained using $R^2 = k(x^{\mu}, x^{\mu})$, for any border support vector x^{μ} (Rajasegarar, et al., 2007).

3.2.2 Self-Organizing Map (SOM)

Competitive neural models such as the self-organizing map (SOM) 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 (Siripanadorn, Hattagam, and Teaumroong, 2010a; 2010b).

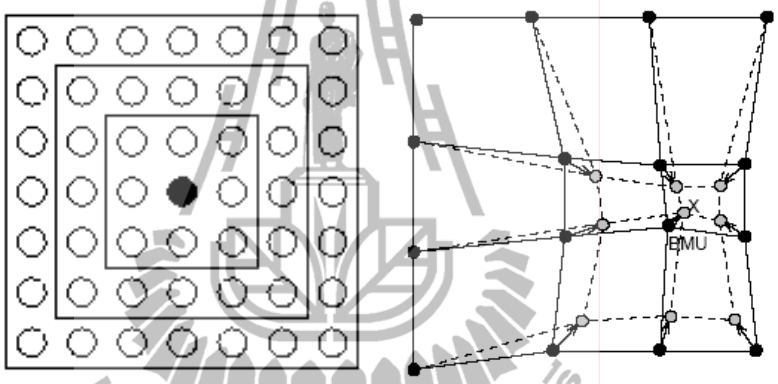


Figure 3.2 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3 neighborhoods, (b) SOM updates the BMU with 1- neighborhood.

The basic SOM consists of a regular grid of map units or neurons as shown in Figure 2 (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 (a) (Siripanadorn, et al., 2010a; 2010b).

Neuron i maintains a weight vector m_i. In order to follow the properties of the input data, such vector is updated during the training process. For example,

Figure 2 (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 $X' = \{x^{\mu} : \mu = 1,2,3,,,s\}$ from the input dataset $X = \{x^{\mu} : \mu = 1,2,3,\ldots,n\}$ is chosen where the number of sample vector X' is s and the number of data vector in a dataset X is n. 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 $m_i(t)$ is the closest to the input vector $x^{\mu}(t)$. We denote such weight vector by $m_c(t)$ and refer to it as the Best-Matching Unit (BMU), which is (Siripanadorn, et al., 2010a; 2010b)

$$\left\| x^{\mu}(t) - m_c(t) \right\| = \arg\min_{\forall i} \left\| x^{\mu}(t) - m_i(t) \right\| \tag{3.5}$$

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 (Siripanadorn, et al., 2010a; 2010b)

$$m_i(t+1) = m_i(t) + \eta_t h_c(i,t)\left[x^{\mu}(t) - m_i(t) \right] \tag{3.6}$$

where t is the iteration index, $x^{\mu}(t)$ 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 (Siripanadorn, et al., 2010a; 2010b)

$$h_c(i,t) = \exp\left[-\frac{\|r_c(t) - r_i(t)\|^2}{2\sigma^2(t)}\right] \tag{3.7}$$

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 (Siripanadorn, et al., 2010a; 2010b).

3.3 Data Compression

3.3.1 Discrete Wavelet Transform (DWT)

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 (Siripanadorn, et al., 2010a; 2010b):

$$a_{j+1}^{DWT}(f) = \sum_{n} h_0(n-2f) a_j^{DWT}(f) \tag{3.8}$$

$$d_{j+1}^{DWT}(f) = \sum_{n} g_0(n-2f) a_j^{DWT}(f) \tag{3.9}$$

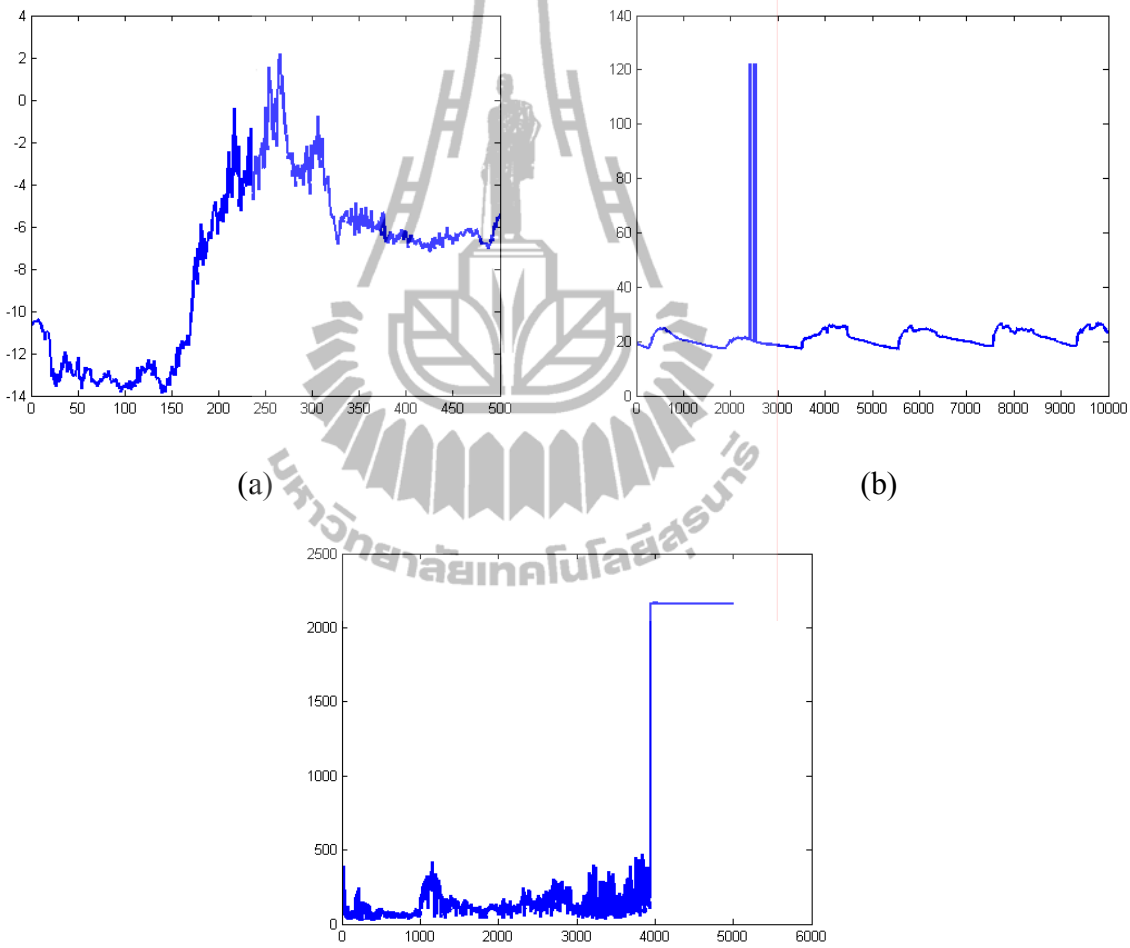
where the rough-scale (or approximation) coefficients a_j^{DWT} are convolved separately with h_0 and g_0, the wavelet function and scaling function, respectively, n is the time scaling index, f is the frequency translation index for wavelet level j. The resulting coefficient is down-sampled by 2. This 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} (Siripanadorn, et al., 2010a; 2010b).

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.

3.4 Datasets for Experiment

We categorized faults into 3 types as shown in Figure 3.3, i.e., noisy faults, short faults and constant faults (Sharma, Golubchik, and Govindan, 2010). A noisy fault is a fault that occurs when variance of the sensor readings increases and affects a number of successive samples. A short fault is a sharp change in the measurement value between two successive data points and affects a single sample at a time. A constant fault is a fault that occurs when a constant value for a large number of successive samples is reported.

We used both synthetic and real world datasets. Three real-world datasets were used for the performance evaluation, namely, INTEL (The INTEL Lab, Online, 2004), SensorScope (The SensorScope Lausanne Urban Canopy Experiment Project: LUCE, Online, 2006), and NAMOS (Network Aquatic Microbial Observing System, Online, 2006) datasets.



(a) (b)

(c)

Figure 3.3 Fault in sensor readings

(a) Noisy faults, (b) Short faults and (c) Constant faults.

3.4.1 Synthetic Data

The synthetic data is generated from a mixture of Gaussian distributions with means randomly selected from (0.3, 0.35, 0.45) and with a standard deviation of 0.03 using MATLAB. Data was generated for 15 sensor nodes and two features of 106 data vectors per sensor node. The combined data comprised 1590 data vectors. Then we introduced a number of faults uniformly distributed ranging between [0.50,1] to each feature of the data (Rajasegarar, Leckie, Palaniswami, and Bezdek, 2007). The amount of faults was represented by the notation a/s, where "a" is the amount of faults per series and "s" is the amount of series of faults, resulting in the total amount of $a \times s$ faults. The generated faults added to the input data ranged from noisy fault which was 20/4, then 10/8, 5/16, and finally to short faults which was 1/80. All these types of faults gave a total of 80 faulty data. The whole dataset was normalized to the range [0, 1]. The exact positions of the faults injected in the input data were predetermined and was later used to detect true and false positive alarms.

3.4.2 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 (The INTEL Lab, Online, 2004). 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. By considering the positions of anomalous data, we found that this dataset had short faults.

3.4.3 SensorScope Station no.39 dataset (SS39)

In this experiment, we presented the results on anomaly detection in one KPI of SensorScope which was collected from weather station no.39 (SS39). Using visual inspection and the histogram method, the lower and upper threshold valued used for anomaly detection were 1.5 and 9. By considering the positions of anomalous data, we found that this dataset depicted short faults.

3.4.4 SensorScope pdg2008-metro-1 dataset (pdg2008)

In the experiment, we used two types (KPIs) of data in the pdg2008-metro-1 dataset for anomaly detection, 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 were -14 and 4 for the surface temperature and -12 and 4 for the ambient temperature. By considering the positions of anomalous data, we found that this dataset contained noisy faults.

3.4.5 NAMOS dataset

In this dataset, 9 buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, for over 24 hours in August, 2006 (Network Aquatic Microbial Observing System, Online, 2006). We analyzed the measurements from chlorophyll sensors on buoys no. 103 for 10^4 samples. 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. By considering the positions of anomalous data, we found that constant faults were present in this dataset.

3.5 Experiment Results

This section consists of two parts. First, we evaluated the performance of the proposed integration of DWT and OCSVM algorithm by detecting anomalies in series of synthetic data and real world datasets. We then proceed to evaluate the performance of a previous technique using DWT and SOM (Siripanadorn, et al., 2010a; 2010b) in comparison to our algorithm.

3.5.1 Evaluation of DWT with OCSVM

In this chapter, we use the linear kernel as the distance based kernel. The linear kernel function for data vectors x^{μ} and x^{ϖ} is given by $k_{linear}(x^{\mu}, x^{\varpi}) = \phi(x^{\mu}) \cdot \phi(x^{\varpi})$.

In each simulation, we recorded the false positives, which occurred when a normal measurement was identified as anomalous by the detector, and the true positives, which occurred when an actual anomalous measurement was correctly identified by the detector. The false positive rate (FPR) was computed as the percentage ratio between the false positives and the actual normal measurements, and the detection rate (DR) was computed as the percentage ratio between the true positives and the actual normal measurements.

In our proposed integration of DWT with OCSVM (OCSVM+DWT) algorithm, we improved the performance of the OCSVM part of the algorithm by replacing the original set of input data with low pass or high pass DWT coefficients by using Haar mother wavelet. The low (high) pass DWT coefficients obtained from DWT were referred as low (high) pass data with just half of the original data size, whereas the original data were referred as uncompressed data.

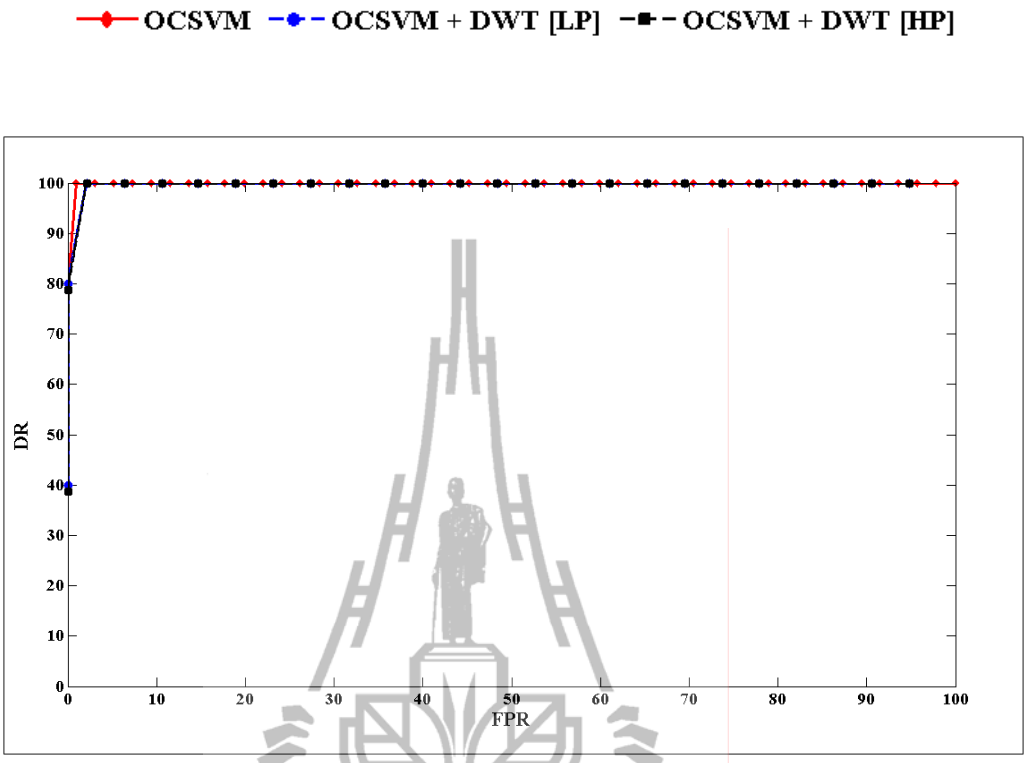


Figure 3.4 ROC for synthetic data inject 1/80 fault

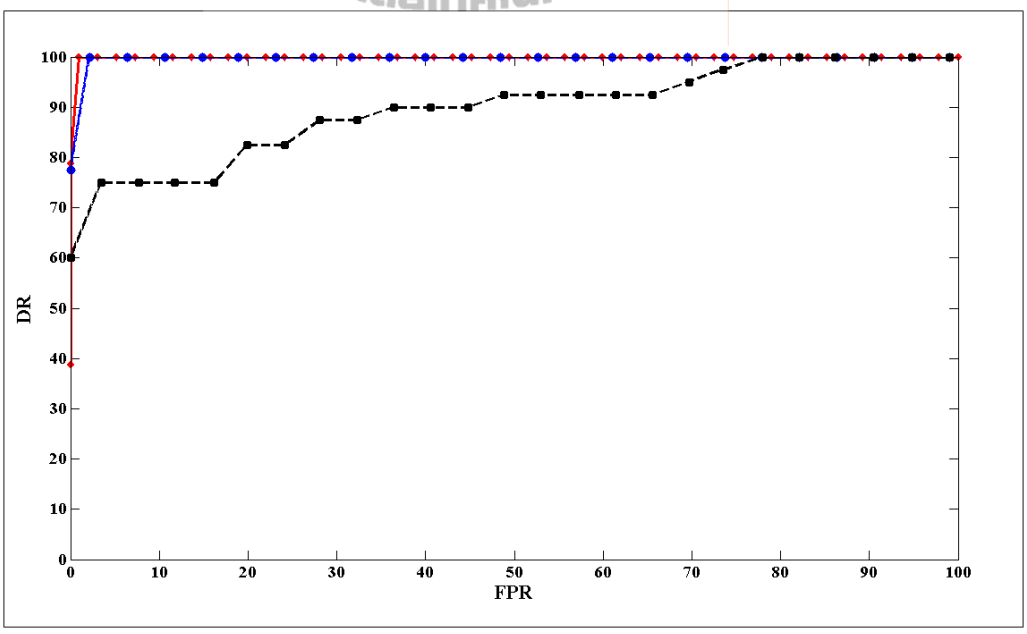


Figure 3.5 ROC for synthetic data inject 5/16 fault

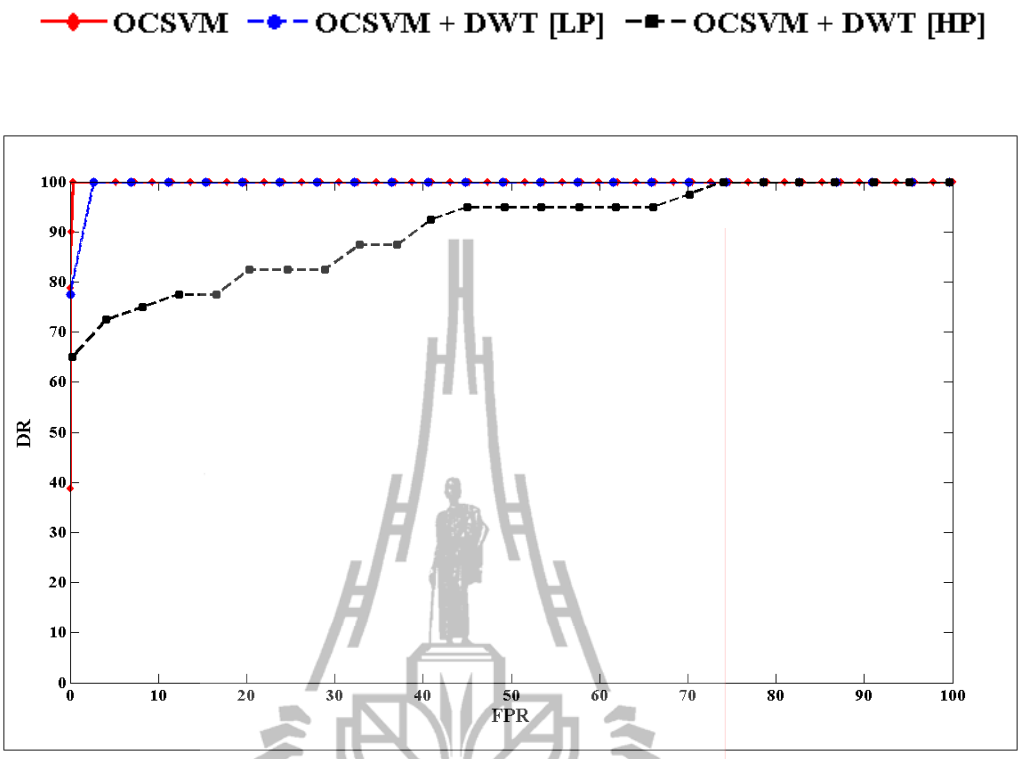


Figure 3.6 ROC for synthetic data inject 10/8 fault

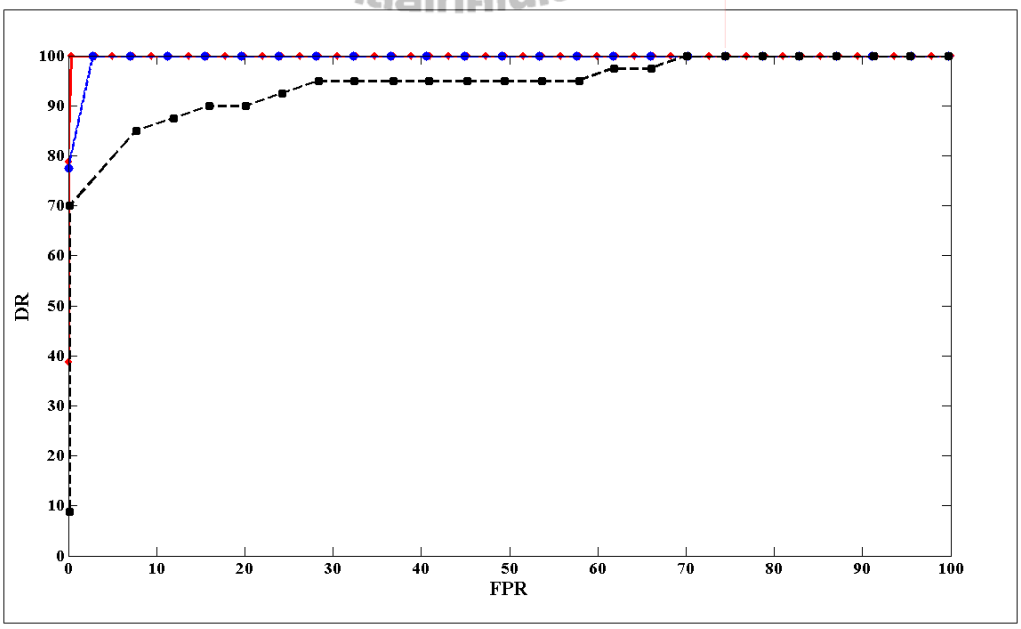


Figure 3.7 ROC for synthetic data inject 20/4 fault

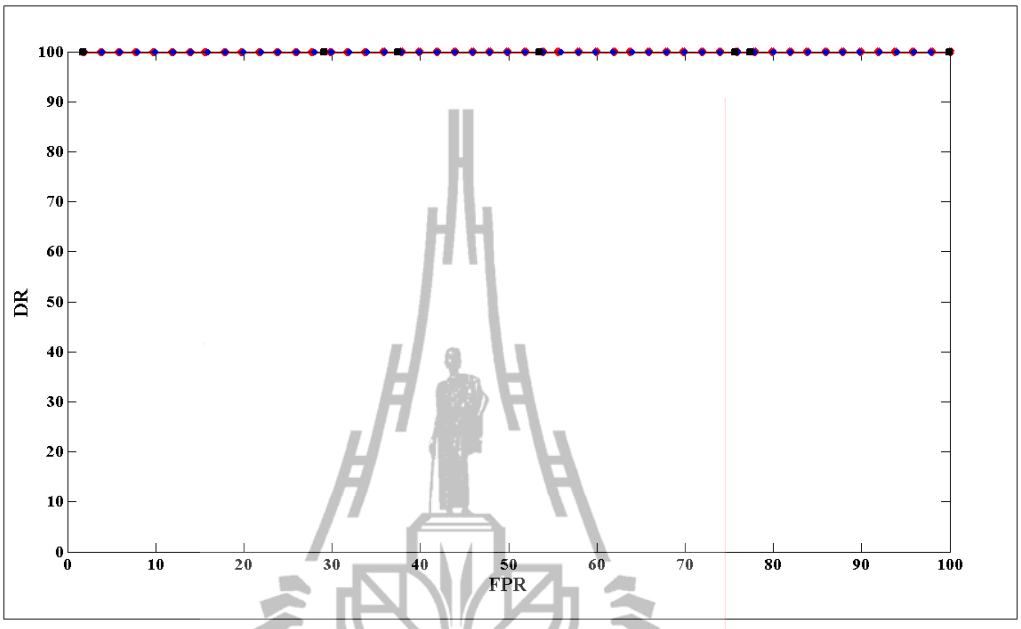


Figure 3.8 ROC for INTEL dataset

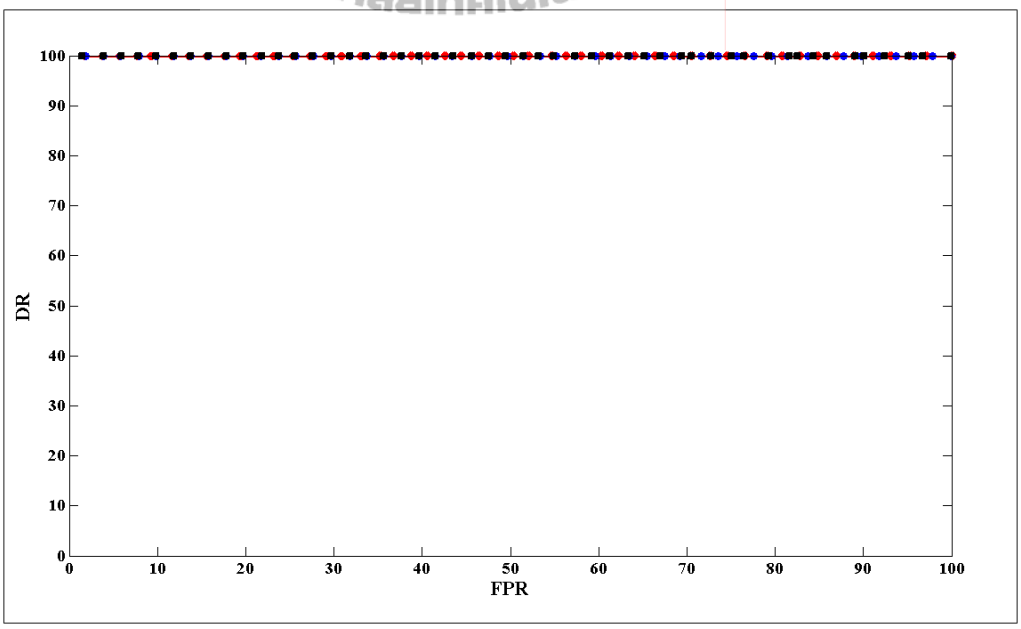


Figure 3.9 ROC for SS39 dataset

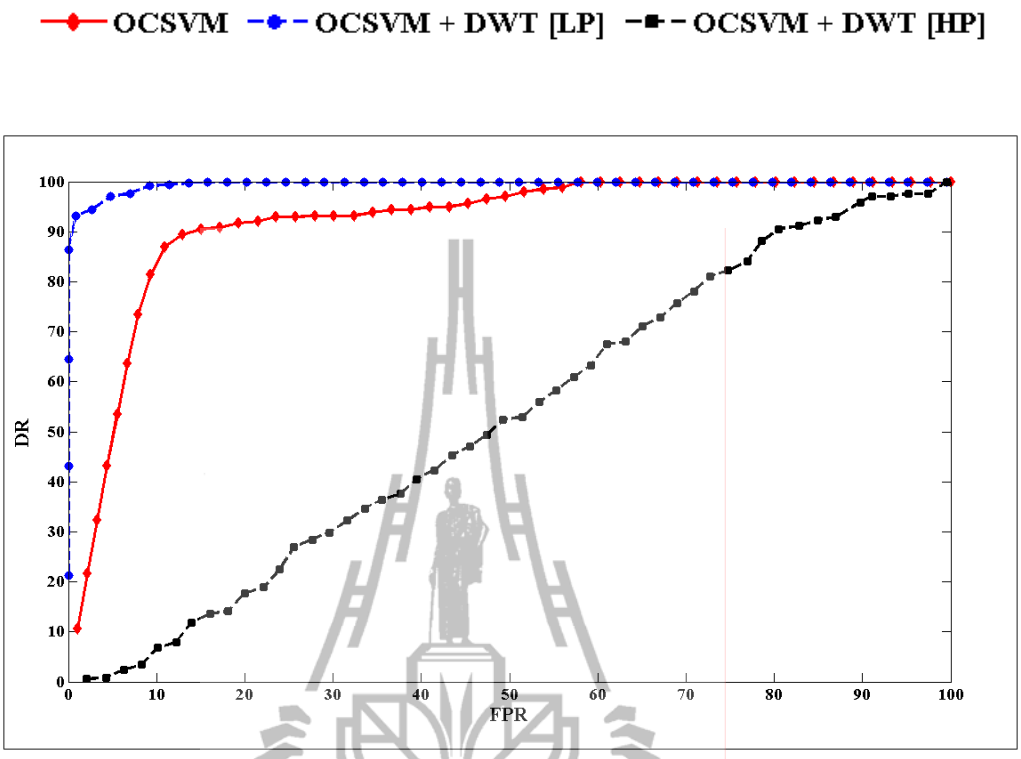


Figure 3.10 ROC for pdg2008-metro-1 dataset

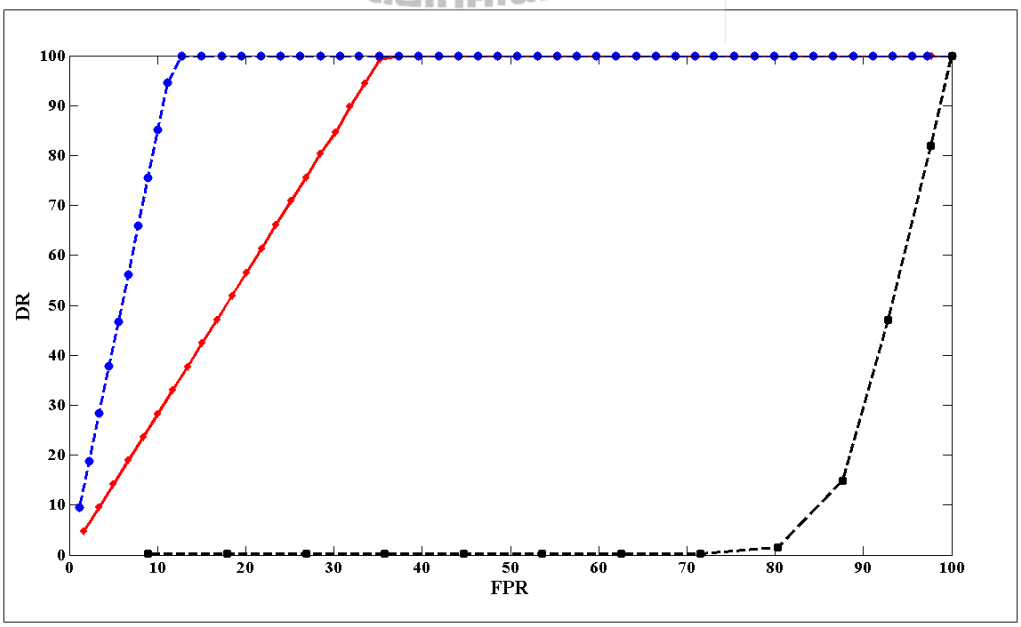


Figure 3.11 ROC for NAMOS dataset.

Figure 3.4 – 3.11 show the receiver operating characteristics (ROC) curve obtained for the OCSVM+DWT schemes for various datasets by varying the ν parameters from 0.02 to 1 in increments of 0.02. The results showed that ν significantly affected DR and FPR in OCSVM. The value ν is the fraction of detected outliers (Y. Zhang et al., 2009), and therefore is directly proportional to the radius R (see Figure 3.1). Hence, the greater ν value, the more the outliers detected (thus the higher DR and FPR).

Figure 3.4 shows the synthetic dataset injected short faults results. Note that all algorithms performed equally well. Figures 3.5, 3.6 and 3.7 show results for the synthetic dataset injected with noise faults. Note that the OCSVM+DWT (LP) performed equally well in terms of DR as the OCSVM alone with uncompressed data. However, the OCSVM+DWT with high pass data gave the worst performance. This was because HP coefficients reflect the rate of changes between two successive samples. Therefore, HP coefficients were more suitable for short faults whereas LP coefficients were more suitable for slower changing faults like noise faults.

Figures 3.8 – 3.11 show the real world dataset results. Figures 3.8 and 3.9 show that the INTEL dataset and the SS39 datasets respectively, gave 100% DR for all algorithms. This was because the INTEL and the SS39 dataset contain short faults which with high amplitude which can be easily detected. Figures 3.10 and 3.11 illustrate the results for the pdg2008-metro-1 and the NAMOS datasets respectively. Note that OCSVM+DWT (LP) obtained higher DR and lower FPR, thereby outperforming the OCSVM alone with uncompressed data and OCSVM+DWT (HP). This was because the pdg2008-metro-1 dataset contained noise faults and the NAMOS dataset contained a constant fault. Both types of faults were trend-like

changes and therefore were more significant when captured with LP coefficients than HP coefficients. Therefore, OCSVM+DWT (HP) data performed worst.

Note that all synthetic data and real world dataset results showed OCSVM+DWT (LP) performed well in terms of %DR. This was because the LP coefficients gave a higher amplitude of faults than the original dataset making it easier to detect anomaly (see Appendix I).

3.5.2 Comparison with previous work

In this section, the best result of the proposed OCSVM+DWT algorithm from each dataset was selected to be compared with the results from the OCSVM alone, SOM alone and the SOM+DWT algorithms in (Siripanadorn, et al., 2010a; 2010b) under the same datasets described previously. For each dataset, we selected the values of ν to use in our proposed algorithm which gave the best performance (the highest DR, the lowest FPR) as follows:

- $\nu = 0.06$ for the all synthetic datasets,
- $\nu = 0.02$ for the INTEL dataset and the SS39 dataset,
- $\nu = 0.14$ for the pdg2008-metro-1 dataset,
- $\nu = 0.22$ for the NAMOS dataset.

The SOM alone and SOM+DWT algorithms were trained with 50 iterations to prevent under-trained conditions using a 50 by 50 neuron grid. The number of samples used to train the SOM alone and the SOM+DWT algorithms were selected from a fault-free region using 2000 samples for INTEL and pdg2008-metro-1 datasets, 3000 samples for NAMOS datasets and 3200 samples for SS39 datasets.

Figure 3.12 shows that using OCSVM alone and the OCSVM+DWT algorithms with synthetic data injected by 1/80 short faults obtained 100% DR,

though OCSVM+DWT obtained marginal FPR of 2.12% and the OCSVM alone 0.99%. The SOM alone and SOM+DWT results were more conservative attaining less DR and no FPR. As the faults became more bursty (more noise faults) in Figures 3.13, 3.14, and 3.15, 100% DR was obtained by OCSVM alone and OCSVM+DWT (LP) whereas OCSVM+DWT (HP) obtained 75-85% DR. However, this was at the expense of an increase in FPR of 0.13-0.99%, 6.36-7.02%, and 7.68-8.21% respectively for OCSVM alone, OCSVM+DWT (LP) and OCSVM+DWT (HP). Note that SOM alone with uncompressed data gave 86-90% DR with no FPR. SOM+DWT (LP) gave 76.4-83.7% DR with no FPR but using just half of the input data.

As for the real world datasets, the INTEL and SS39 datasets contained only short faults and were therefore easy to detect. Results in Figures 3.16 and 3.17 agree with Figure 3.12 with all OCSVM based algorithms obtaining 100% DR but FPR up to 1.9%. On the other hand, the SOM based algorithms obtained 100% DR but with FPR slightly lower of up to 1.09%. The improved performance for all algorithms was possibly due to the fact that the short faults in the INTEL and SS39 datasets were detected more easily than the synthetic dataset.

Figure 3.18 depicts results for the pdg2008 dataset which contained noise faults. Note that with the presence of noise faults, FPR for OCSVM based algorithms was greater than the SOM based algorithms which agreed with results in the synthetic dataset in Figures 3.13, 3.14, and 3.15. However, OCSVM+DWT (LP) gave the best results obtaining 97.04% DR with 4.81% FPR, thereby outperforming OCSVM alone.

Figure 3.19 illustrates the results for the NAMOS dataset which comprised of constant faults. Such faults were difficult to detect since they appear as

normal data. Even with SOM alone and SOM+DWT (LP) can fail to detect such faults if under-trained (Siripanadorn, et al., 2010a; 2010b). Note that SOM alone, SOM+DWT (LP) near 100% DR while OCSVM+DWT (LP) attained 100%, though the FPR was 12.77% for OCSVM+DWT (LP) but negligible FPR for SOM alone, SOM+DWT (LP). These results suggest that with data compression and using just half of the data input, OCSVM+DWT (LP) algorithm is suited for short and noise faults whereas SOM+DWT (LP) is suited for short and constant faults.

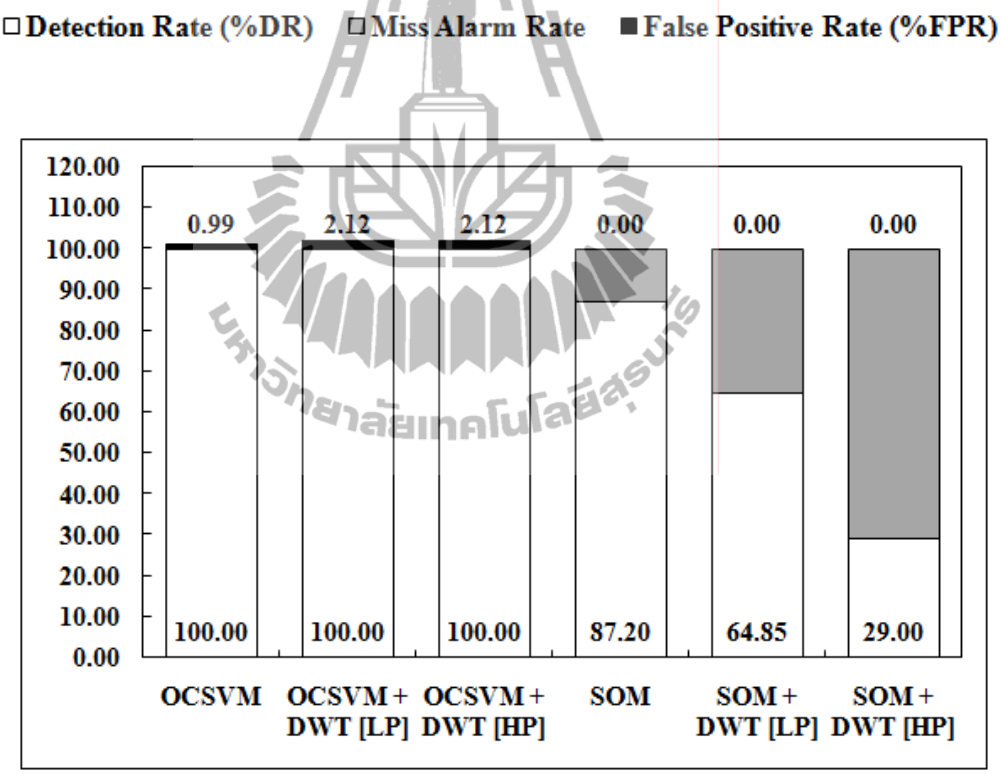


Figure 3.12 Detection Rate with different algorithm for 1/80 fault synthetic data.

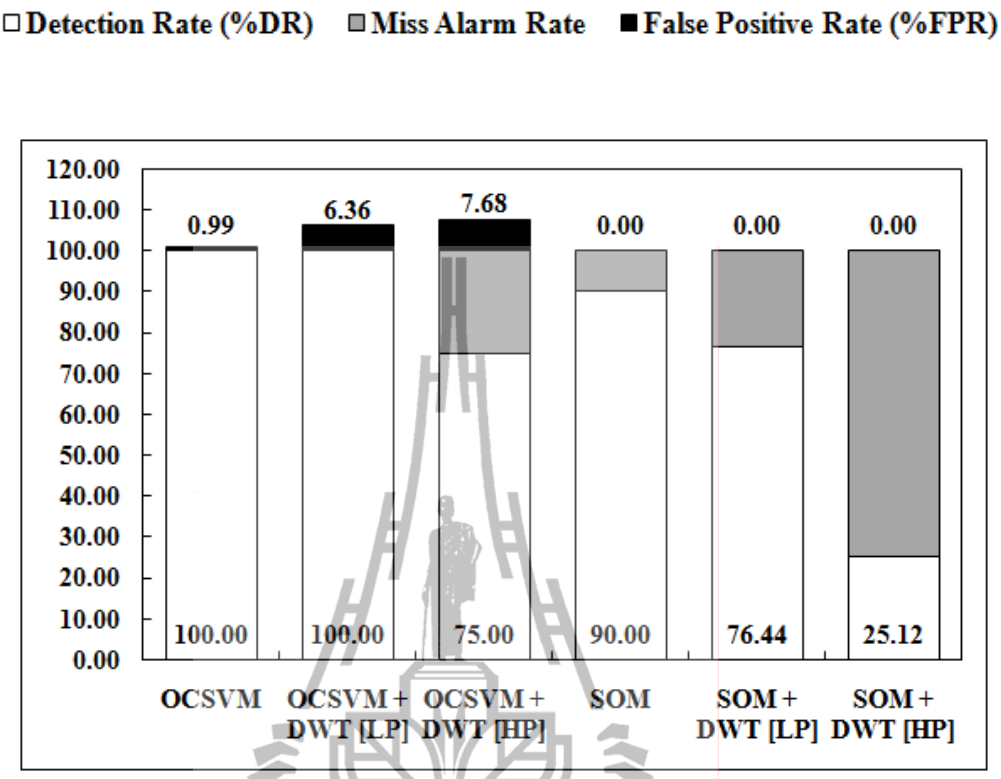


Figure 3.13 Detection Rate with different algorithm for 5/16 fault synthetic data.

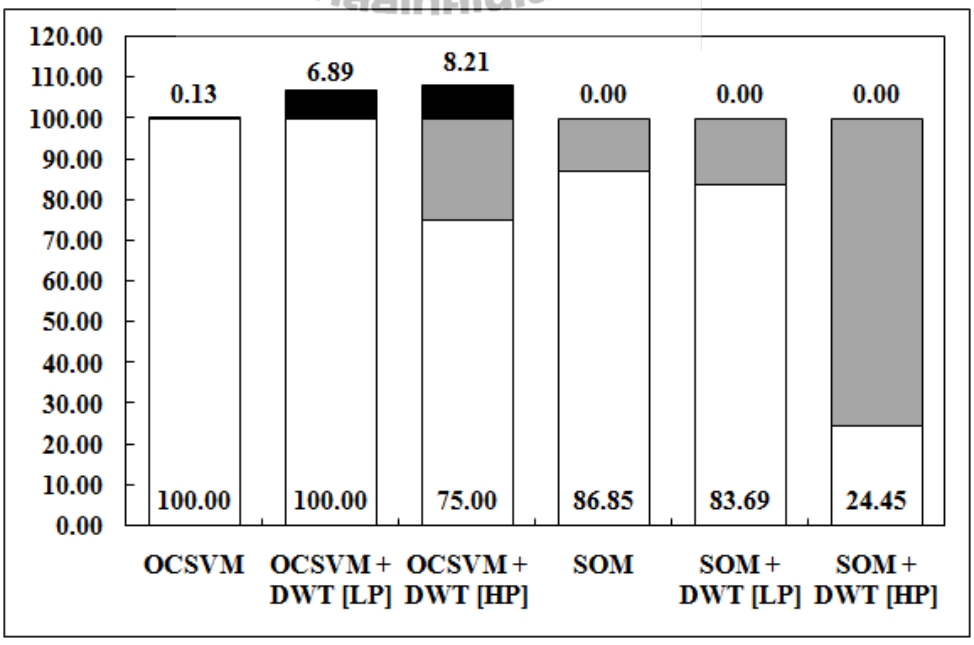


Figure 3.14 Detection Rate with different algorithm for 10/8 fault synthetic data.

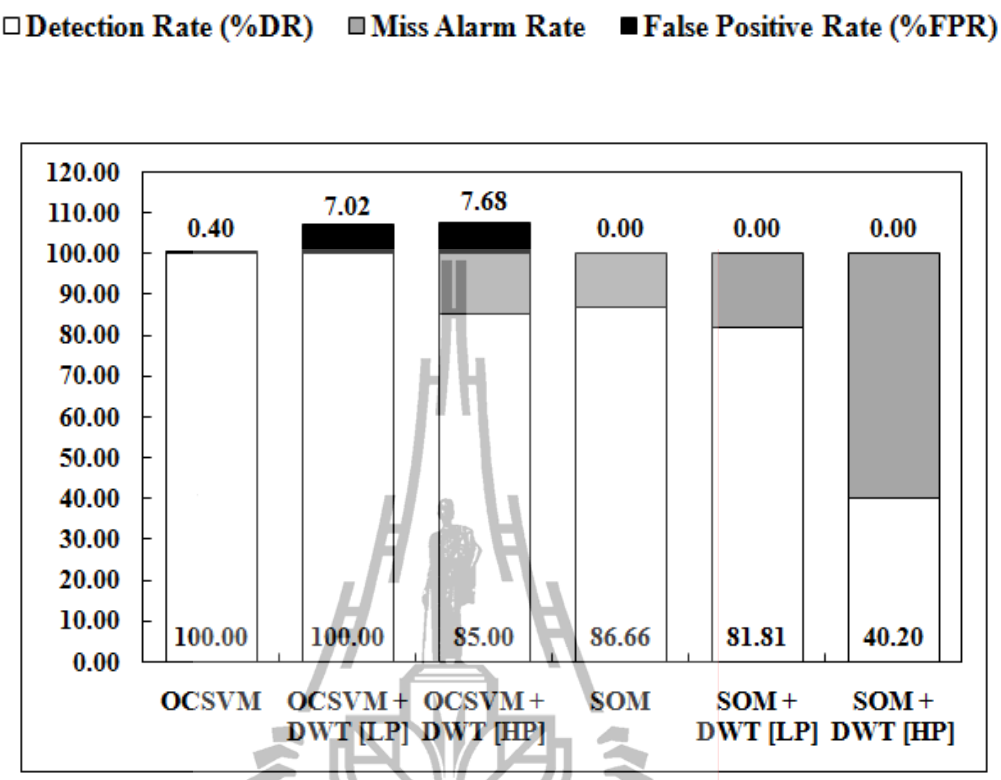


Figure 3.15 Detection Rate with different algorithm for 20/4 fault synthetic data.

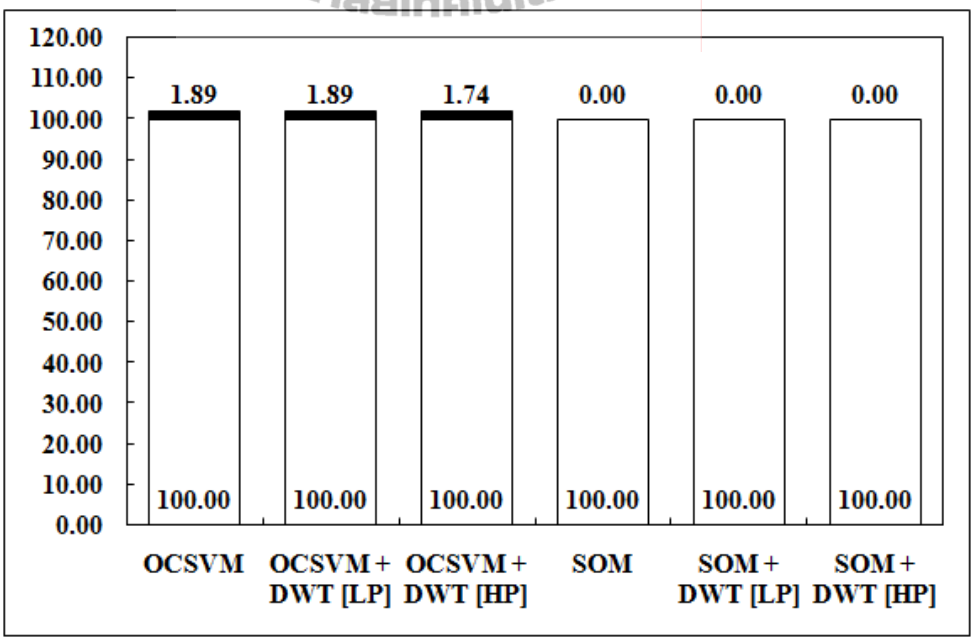


Figure 3.16 Detection Rate with different algorithm for the INTEL dataset.

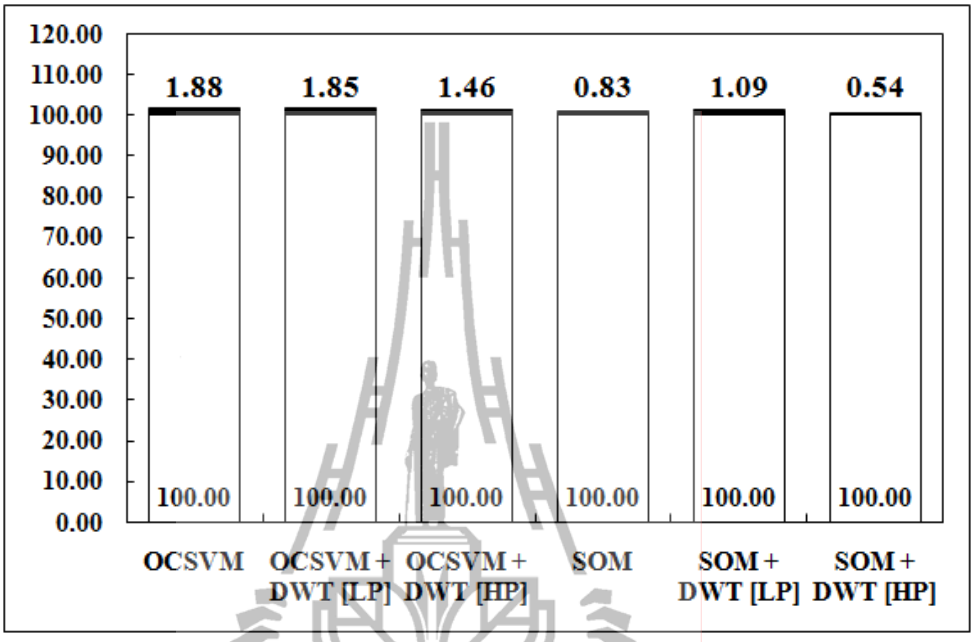


Figure 3.17 Detection Rate with different algorithm for the SS39 dataset.

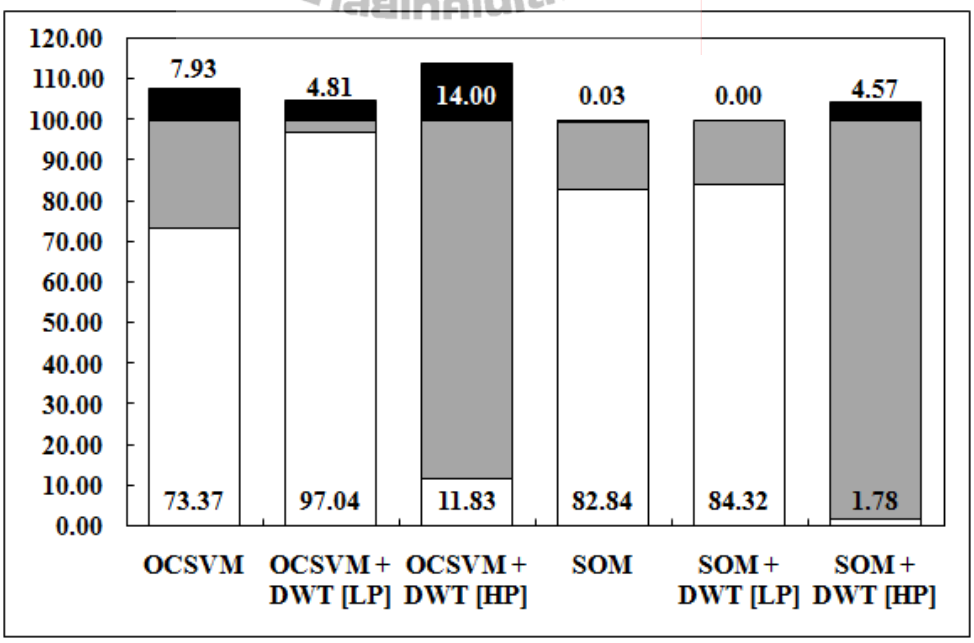


Figure 3.18 Detection Rate with different algorithm for the pdg2008 dataset.

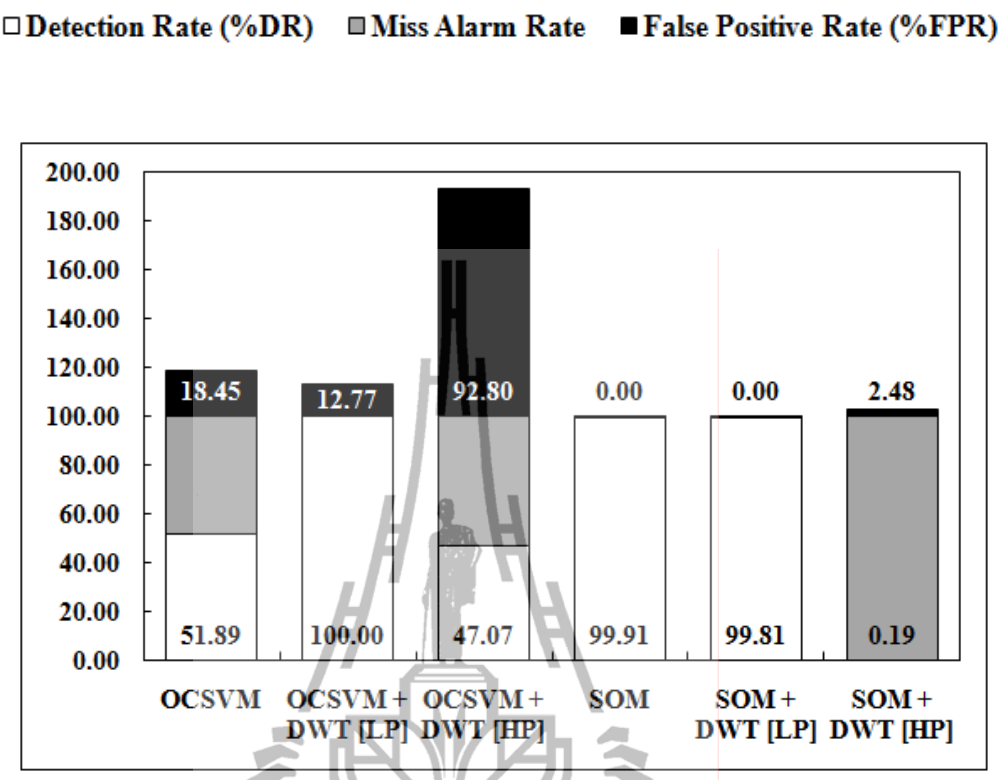


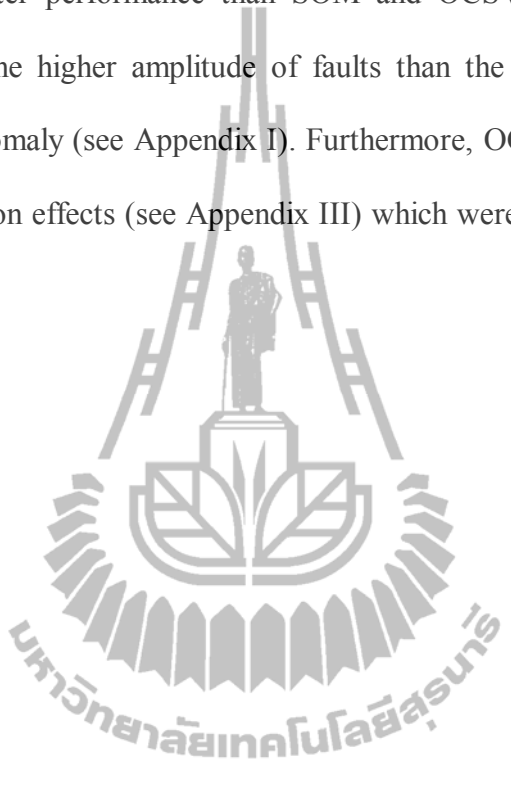
Figure 3.19 Detection Rate with different algorithm for the NAMOS dataset.

3.6 Conclusion

We proposed the integration of OCSVM and DWT for anomaly detection in WSNs. We numerically evaluated the algorithm using MATLAB and tested it with both synthetic data and real world datasets. For synthetic data, our proposed algorithm with LP coefficients achieved 100% DR with marginal increase in FPR when compared with all other algorithms. For real world datasets, our proposed algorithm performed best by achieving nearly 100% DR although with slightly higher FPR for datasets containing short and noise faults. These results suggest that with data compression and using just half of the data input, OCSVM+DWT (LP) algorithm is

suited for short and noise faults whereas SOM+DWT (LP) is suited for short and constant faults.

Note that, the anomaly detection using SOM and OCSVM with LP coefficient of DWT gave better performance than SOM and OCSVM alone because the LP coefficients gave the higher amplitude of faults than the original dataset making it easier to detect anomaly (see Appendix I). Furthermore, OCSVM was more robust to dataset normalization effects (see Appendix III) which were a motivation for its use in the next chapter.



CHAPTER IV

LIFTING WAVELET TRANSFORM AND ONE-CLASS SUPPORT VECTOR MACHINES FOR ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS

This chapter proposes an integrated data compression and anomaly detection algorithm in WSNs. The contribution of this chapter centers on data compression by using the lifting wavelet transform (LWT) then feeding it to anomaly detection called the one-class support vector machine (OCSVM). We tested our algorithm with several synthetic and real world datasets and compared it with other existing data compression schemes such as principal component analysis (PCA) and discrete wavelet transform (DWT).

4.1 Introduction

Our previous work in chapter III showed that the integration of anomaly detection by the one-class support vector machine (OCSVM) and data compression by DWT outperformed existing approaches. In addition, Kiziloren and Germen (2009) proposed integration between anomaly detection and data compression using the principal component analysis (PCA), which is a suitable tool for reducing the dimension of the datasets prior to feeding data to an anomaly detection algorithm. These approaches can increase the efficiency of anomaly detection. Motivated by their findings, we extend our work in chapter III to integrate OCSVM anomaly detection

with alternative data compression.

The Principal Component Analysis (PCA) reduces the dimension of the datasets by reducing the variates or number of parameter types measured. However, it is possible that certain features of the dataset may be lost as a result because PCA can ignore some components which have lesser significance (Smith, 2002). This is likely to decrease the anomaly detection efficiency.

On the other hand, the Discrete Wavelet Transform (DWT) has the capability to encode the finer resolution of the original time series with its hierarchical coefficients. DWT reduces only the size of the data vector, it does not reduce the parameter variates. Furthermore, DWT can be computed efficiently in linear time, which is important while dealing with large datasets (Siripanadorn, et al., 2010). However, (Acharya and Chakrabarti, 2006; Manjunath and Ravikumar, 2010) proposed that DWT which has been implemented by convolutions, require a larger storage, a larger number of arithmetic computation and higher computational time than a newer wavelet transform which was proposed in 1998 by Sweldens, called the lifting wavelet transforms (LWT). To the best of our knowledge, the integration between OCSVM and LWT has not yet been proposed. Therefore, the underlying aim of this chapter is to study the effect and efficiency of data compression by using LWT on the OCSVM anomaly detection technique, we then assess its suitability for deployment in resource-constrained conditions in WSNs in comparison with other existing combinations of data compression and anomaly detection schemes.

4.2 Anomaly Detection

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

$$x^{\mu} = \begin{bmatrix} x_1^{\mu} \\ x_2^{\mu} \\ x_3^{\mu} \\ \vdots \\ x_p^{\mu} \end{bmatrix} = \begin{bmatrix} KPI_1^{\mu} \\ KPI_2^{\mu} \\ KPI_3^{\mu} \\ \vdots \\ KPI_p^{\mu} \end{bmatrix} \tag{4.1}$$

where $\mu = 1,2,3,...,n$ is the observation index,

n is the number of data vectors in the dataset,

p is the number of parameter types or key performance indices (KPIs) chosen to monitor the environmental condition.

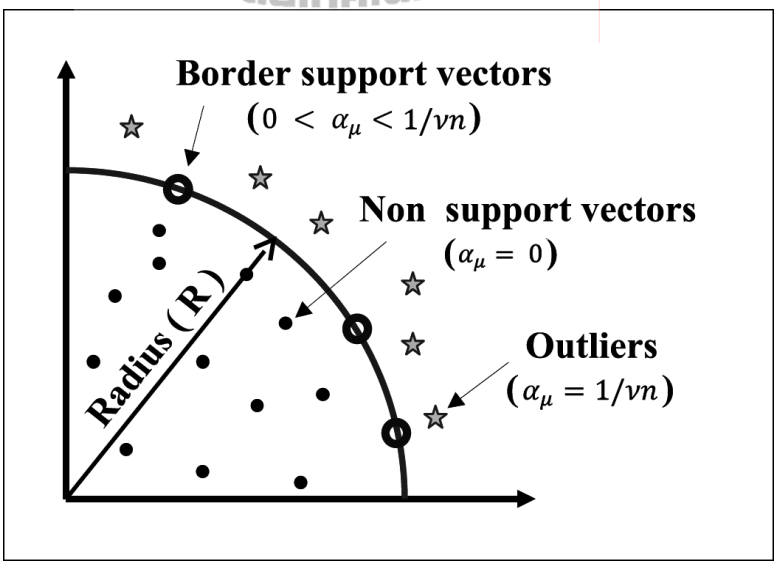


Figure 4.1 Geometry of the quarter-sphere OCSVM

(Rajasegarar, Leckie, Bezdek, and Palaniswami, 2007)

4.2.1 One-Class Support Vector Machines (OCSVM)

In 2004, Tax and Duin have proposed a one-class support vector machines (OCSVM) formulation for outlier detection. Later, Laskov, Schafer, and Kotenko extended the OCSVM to one-side non-negative data which is to require the center of the fitted sphere be fixed at the origin called Quarter-Sphere OCSVM. The geometry of this approach is shown in Figure 4.1

Consider an input dataset $X = \{x^\mu : \mu = 1,2,3,...,n\}$ of p variates data vector $x^\mu = \left[x_1^\mu, x_2^\mu, ..., x_p^\mu\right]$ in the input space $\Re^p$ where the number of data vectors in a dataset X is n. In principle, X is mapped to a feature space $\Re^q$ via a nonlinear function $\phi(\cdot)$, resulting in a set of the image vectors $X_\phi = \left\{\phi(x^\mu) : \mu = 1,2,3,...,n\right\}$ where $\phi(x^\mu) = \left\{\phi(x_1^\mu), \phi(x_2^\mu), \phi(x_3^\mu), ..., \phi(x_q^\mu)\right\}$ is a row vector of image vector. The normal data can be concisely described by a quarter-sphere in Figure 4.1. The aim is to fit a hypersphere in a feature space with minimum effective radius $R > 0$ centered at the origin, encompassing a majority of the image vectors X_ϕ. The presence of anomalies in the data can be treated by introducing slack variables ξ_μ. Mathematically, the problem of fitting the quarter-sphere over the data is described as follows (Laskov, Schafer, and Kotenko, 2004):

$$\min_{R \in \Re^+, \xi \in \Re} \; R^2 + \frac{1}{\nu n} \sum_{\mu=1}^{n} \xi_\mu$$

Subject to: $k(x^\mu, x^\mu) \le R^2 + \xi_\mu$, (4.2)

$$\xi_\mu \geq 0$$

where ξ_μ are the slack variables that allow some of the image vectors to lie outside the sphere. The parameter $\nu \in (0,1)$ is the regularization parameter which controls the fraction of image vectors that lie outside the sphere, i.e., the fraction of image vectors that can be treated as anomalies.

Note that $\left\|\phi(x^\mu)\right\| = \phi(x^\mu)\cdot\phi(x^\mu)^T = k(x^\mu, x^\mu)$ for a Mercer kernel and $k(x^\mu, x^\mu)$ is a kernel function which was used to compute the similarity of any two vectors in the feature space using the original attribute set. The primal problem (4.2) cannot be directly solved (Laskov, et al., 2004). Therefore, the solution is sought to the dual problem as follows (Laskov, et al., 2004; Rajasegarar, Leckie, Bezdek, and Palaniswami, 2007):

$$\min_{\alpha\in\Re} -\sum_{\mu=1}^{n}\alpha_\mu k(x^\mu, x^\mu)$$

Subject to: $$\sum_{\mu=1}^{n}\alpha_\mu = 1 \qquad (4.3)$$

$$0 \leq \alpha_\mu \leq \frac{1}{\nu n}$$

This dual problem in (4.3) is a linear optimization problem, therefore the Lagrangian multiplier $\{\alpha_\mu\}$ can be obtained by using widely available linear optimization techniques. The parameter was used to classified the image vectors

$\{\phi(x^\mu)\}$ as follows (See Figure 1). The image vectors with $\alpha_\mu = 0$ will fall inside the sphere. The image vectors with $0 < \alpha_\mu < \frac{1}{\nu n}$ will reside on the surface of the sphere, and hence are called the border support vectors. Support vectors with $\alpha_\mu = \frac{1}{\nu n}$ are termed as outliers, which fall outside the sphere. Moreover, the radius of the sphere *R* can be obtained using $R^2 = k(x^\mu, x^\mu)$, for any border support vector x^μ (Rajasegarar, et al., 2007).

Furthermore, the solution to the dual problem in (4.3) is affected by the norms of the non-linear mapping of data vectors using kernels $k(x^\mu, x^\mu)$. This created the problem for the application of this solution with the distance-based kernel, as the norms of the kernel are equal for all data vectors and no meaningful solution to the dual problem. In order to solve this problem, the image vectors in the feature space are centered in the space by subtracting the mean from all image vectors (Laskov, et al., 2004; Rajasegarar, et al., 2007):

$$\tilde{\phi}(x^\mu) = \phi(x^\mu) - \frac{1}{n}\sum_{\mu=1}^{n}\phi(x^\mu) \tag{4.4}$$

The dot product of the centered image vector $\tilde{\phi}(x^\mu)$ can be easily computed by using a center kernel matrix K_c as follows (Laskov, et al., 2004; Rajasegarar, et al., 2007):

$$K_c = K - 1_n K - K\,1_n + 1_n K\,1_n \tag{4.5}$$

where 1_n is an $n \times n$ metric with all values equal to $\frac{1}{n}$, K is an $n \times n$ kernel metric consisting of $k(x^{\mu}, x^{\varpi})$ where $\mu, \varpi = 1, 2, 3, ..., n$. If $\mu = \varpi$, we have that $k(x^{\mu}, x^{\varpi}) = k(x^{\mu}, x^{\mu})$ and thus obtain the norm of image vector $\phi(x^{\mu})$. Otherwise, $k(x^{\mu}, x^{\varpi})$ can be obtained from a kernel function, such as linear, polynomial or RBF kernel. Once the image vectors are centered, the norms of the kernels are no longer equal. Hence the dual problem (4.3) can now be easily solved (Laskov, et al., 2004; Rajasegarar, et al., 2007).

4.3 Data Compression

4.3.1 Principal Component Analysis (PCA)

The principal component analysis (PCA) was proposed as a tool for reducing the dimensionality of a dataset (Kiziloren and Germen, 2009). PCA is a classical statistical method which is completely reversible (the original data may be recovered exactly from the PCs), making it a versatile tool, useful for data reduction, noise rejection, visualization and data compression. (Dwinnell, Online, 2010). However, PCA allows ignoring some components of data which has lesser significance by considering the eigenvalues of the covariance matrix of the data. If small eigenvalues are ignored, less information is lost (Smith, Online, 2002).

The PCA framework in Figure 4.2 can be described as follows. First, we find the mean of the input dataset X and subtract it from X, resulting in the adjusted data. Then, we find the covariance matrix of the adjusted data, in order to find its eigenvectors and eigenvalues, respectively. After that, we selected the maximum eigenvalues to obtain the first PCs. The next orders of PCs are selected from the next highest eigenvalues.

Then, the transposed PCs (called feature vector) are multiplied with the adjusted data. At the end, we obtain the compressed data with the same number of data but fewer KPIs than the original dataset X.

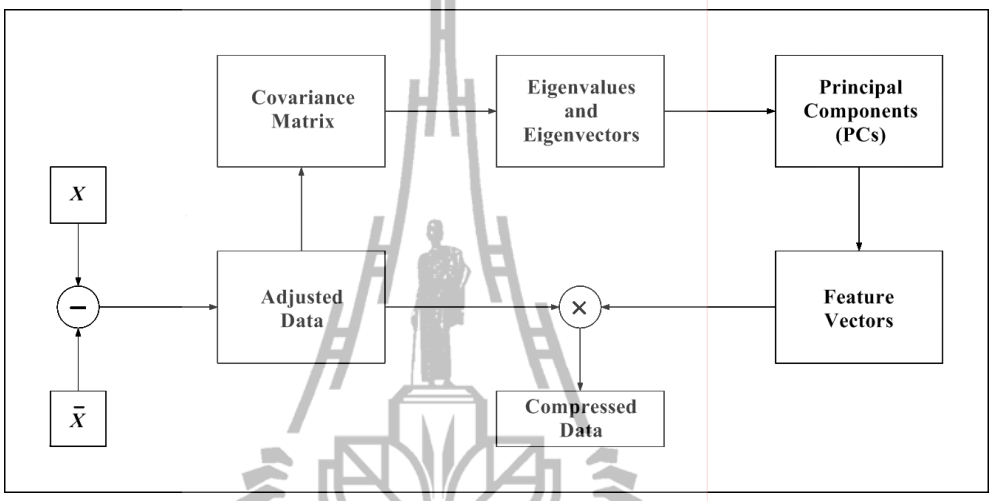


Figure 4.2 Principal Component Analysis (PCA) frameworks.

4.3.2 Discrete Wavelet Transform (DWT)

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 by using the convolution (filtering) based approach. Its major advantage is the multi-resolution representation and time-frequency localization property for signals (Siripanadorn, et al., 2010). 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 (Siripanadorn, et al., 2010):

$$a_{j+1}^{DWT}(f) = \sum_{n} h_0(n-2f) \cdot a_j^{DWT}(f) \tag{4.6}$$

$$d_{j+1}^{DWT}(f) = \sum_{n} g_0(n-2f) \cdot a_j^{DWT}(f) \tag{4.7}$$

where the rough-scale (or approximation) coefficients a_j^{DWT} are convolved separately with the wavelet function h_0 and the scaling function g_0, n is the time scaling index, f 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}.

For data compression, DWT decreases the number of data by half while maintaining the same number of KPIs. 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. However, DWT which has been implemented by convolutions requires a larger number of arithmetic computation and larger storage than the lifting wavelet transforms (LWT) (Achaya and Chakrabarti, 2006).

4.3.3 Lifting Wavelet Transform (LWT)

The Lifting Wavelet Transform (LWT) was proposed as the second generation wavelets (Sweldens, 1998). LWT inherits the multi-resolution characteristics of the first generation wavelets and its main feature is to convert the convolution implementation of DWT into band matrix multiplication by in-place computation. Therefore, LWT requires fewer computations and memory space than

DWT (Sweldens, 1998; Achaya and Chakrabarti, 2006; X. L. Li, J. W. Zhang, and W. H. Fang, 2009).

As shown in Figure 4.3, the LWT algorithm divides the input dataset $X = \{x^{\mu} : \mu = 1,2,3,...,n\}$ or current rough-scale (or approximation) coefficients a_j^{LWT} into 3 stages as follows:

1) Split: the input dataset or current rough-scale (or approximation) coefficients a_j^{LWT} is split into the even half $a_{even\_j}^{LWT}$ and the odd half $a_{odd\_j}^{LWT}$.

2) Predict: the odd half is predicted by subtracting the linear combination of even half from the odd half resulting in the prediction error or high pass coefficients or detail coefficients d_{j+1}^{DWT}.

3) Update: the even half is updated by adding them to a linear combination of the prediction error resulting in low pass or approximation coefficients a_{j+1}^{DWT}.

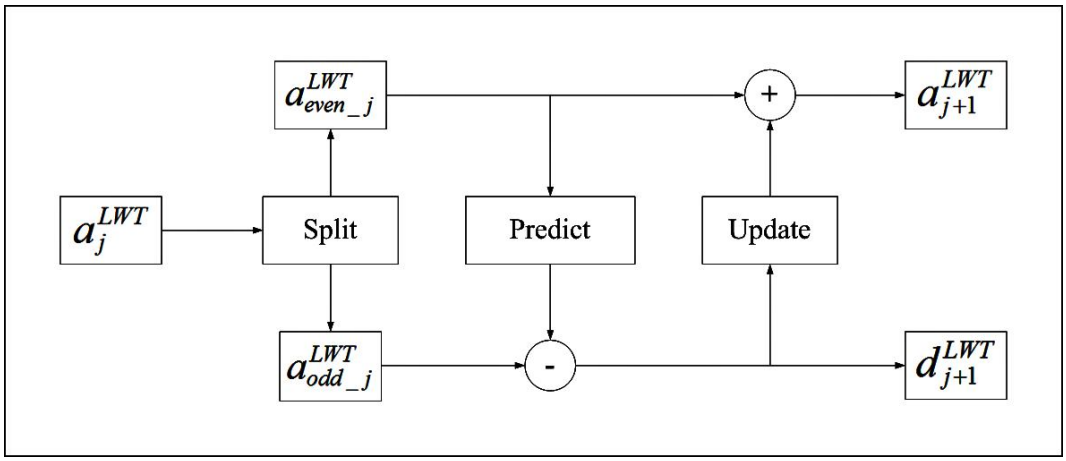


Figure 4.3 Lifting Wavelet Transform (LWT) frameworks.

Note that LWT uses linear combination to find the wavelet coefficients while DWT uses the convolution, indicating that LWT is less computationally intensive than DWT (Manjunath and Ravikumar, 2010).

4.4 Experiment Results

In this section, we used MATLAB to numerically evaluate the performance of four data compression algorithms namely, our proposed OCSVM+LWT, and compared it with OCSVM alone (uncompressed data), OCSVM+PCA (Kiziloren and Germen, 2009) and OCSVM+DWT (Takianngam, et al., 2011), by detecting anomalies in series of synthetic data and real world datasets.

We used the linear kernel as the distance based kernel. The linear kernel function for data vectors x^{μ} and x^{ϖ} is given by $k_{linear}(x^{\mu}, x^{\varpi}) = \phi(x^{\mu}) \cdot \phi(x^{\varpi})$

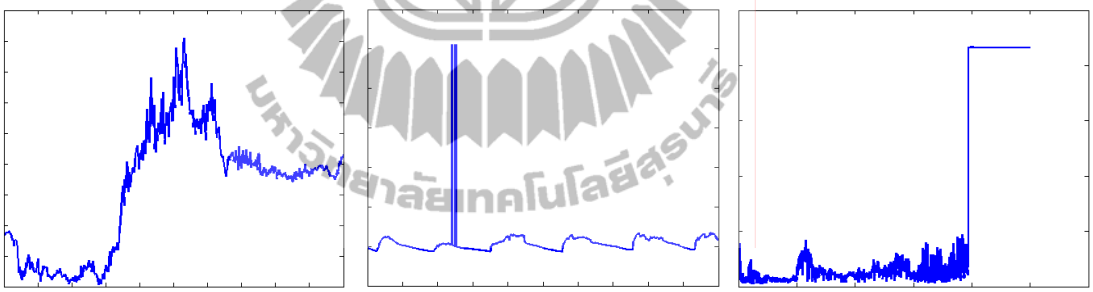
In each simulation, we varied the ν parameter in (4.3) from 0.02 to 1 in increments of 0.02. After that, we recorded the false positives, which occurred when a normal measurement was identified as anomalous by the detector, and the true positives, which occurred when an actual anomalous measurement was correctly identified by the detector. The false positive rate (FPR) was computed as the percentage ratio between the false positives and the actual normal measurements, and the detection rate (DR) was computed as the percentage ratio between the true positives and the actual normal measurements. Finally, we showed the results in the Receiver Operating Characteristics (ROC) curve.

4.4.1 Datasets for Experiment

We categorized faults into 3 types as shown in Figure 4.4, i.e., noise, short and constant faults (Sharma, Golubchik, and Govindan, 2010). A noise fault is a

fault that occurs when the variance of the sensor readings increases and affects a number of successive samples. A short fault is a sharp change in the measurement value between two successive data points and affects a single sample at a time. A constant fault is a fault that occurs when a constant value for a large number of successive samples is reported.

We studied both synthetic and real world datasets. Three real-world datasets were used for the performance evaluation, namely, INTEL (The Intel Lab, Online, 2004), SensorScope pdg2008-metro-1 (The SensorScope Lausanne Urban Canopy Experiment Project (LUCE), Online, 2006), and NAMOS (Network Aquatic Microbial Observing System, Online, 2006) datasets.



(a) (b) (c)

Figure 4.4 Faults in sensor readings

(a) noise faults, (b) short faults and (c) constant faults.

4.4.1.1 Synthetic Data

The synthetic data in Figure 4.5 – 4.10 were generated from a mixture of Gaussian distributions with means randomly selected from (0.3, 0.35, 0.45) and with a standard deviation of 0.03 using MATLAB. Data were generated for 15 sensor nodes and two KPIs of 106 data vectors per sensor node. The combined data

comprised 1590 data vectors. Then we introduced a number of faults uniformly distributed ranging between [0.50, 1] to each KPI of the data. The amount of faults was represented by the notation a/s, where “a” is the amount of faults per series and “s” is the amount of series of faults, resulting in the total amount of a×s faults. The generated faults added to the input data ranged from constant fault (80/1), then noisy faults (20/4), and finally short faults (1/80). All these types of faults gave a total of 80 faulty data. The whole dataset was normalized to the range [0, 1]. The exact positions of the faults injected in the input data were predetermined and later used to detect true and false positive alarms.

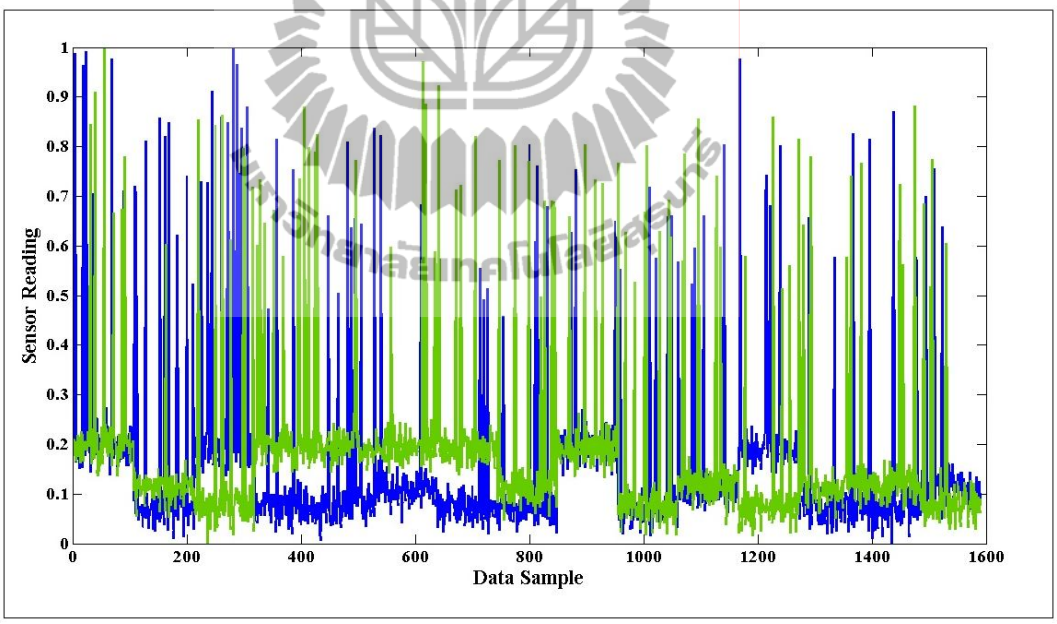


Figure 4.5 2KPI Synthetic data with 1/80 faults.

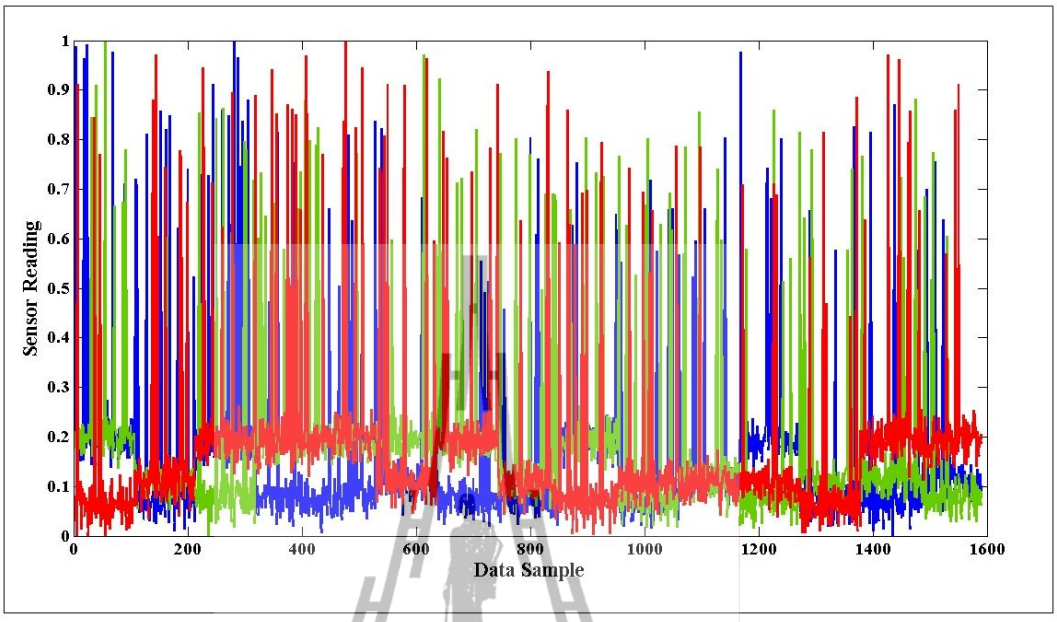


Figure 4.6 3KPI Synthetic data with 1/80 faults.

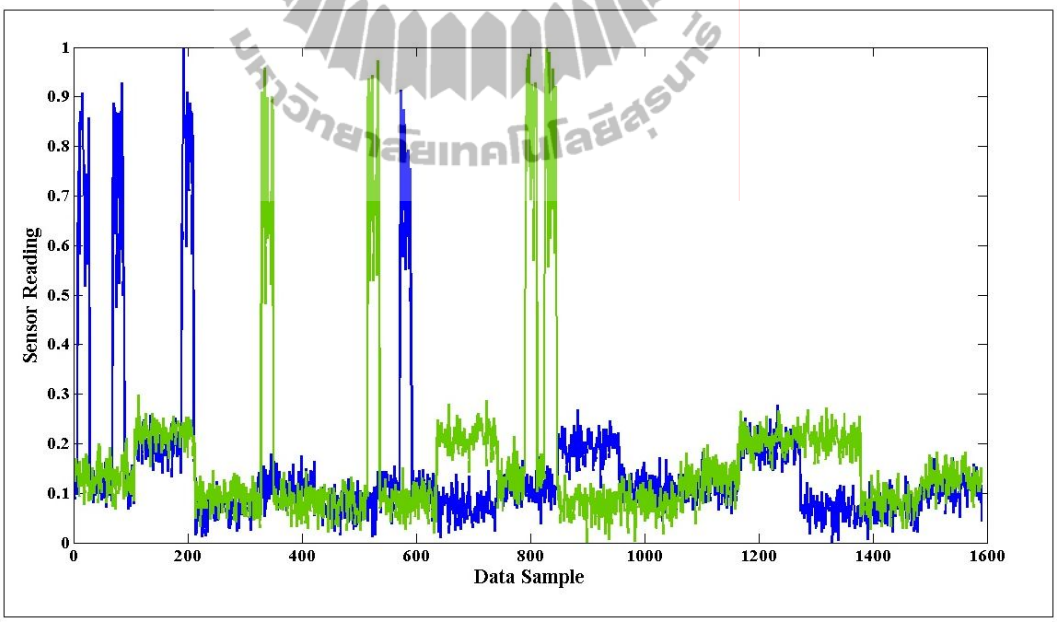


Figure 4.7 2KPI Synthetic data with 20/4 faults.

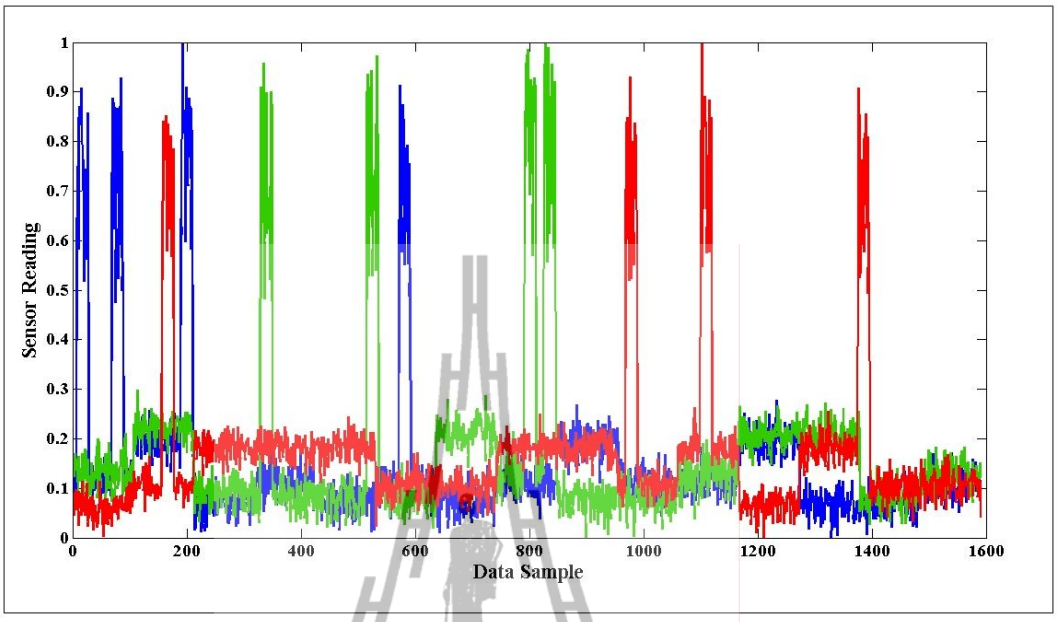


Figure 4.8 3KPI Synthetic data with 20/4 faults.

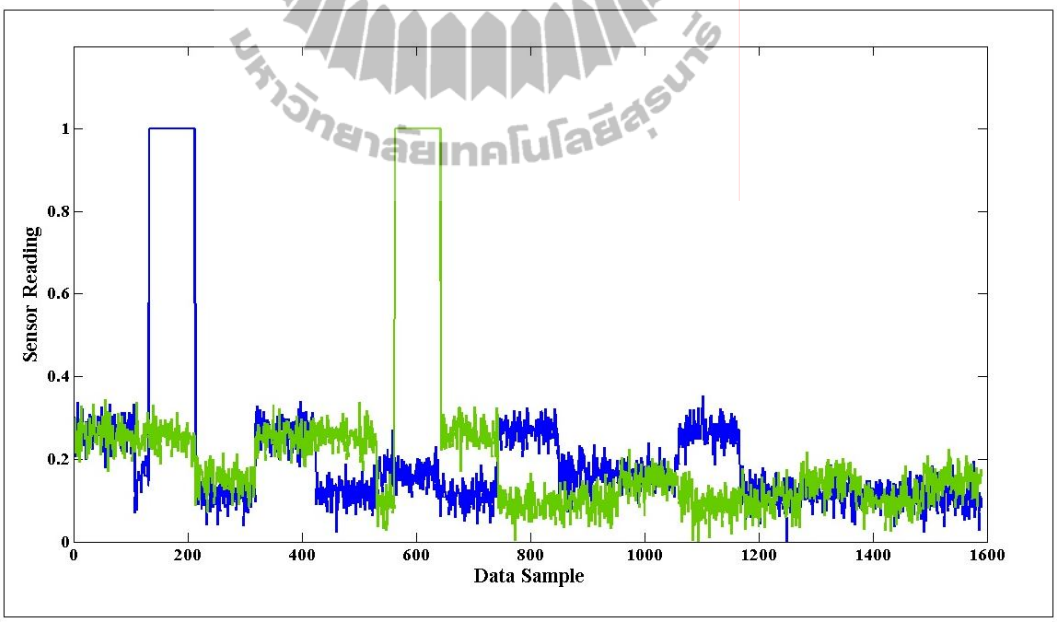


Figure 4.9 2KPI Synthetic data with 80/1 fault.

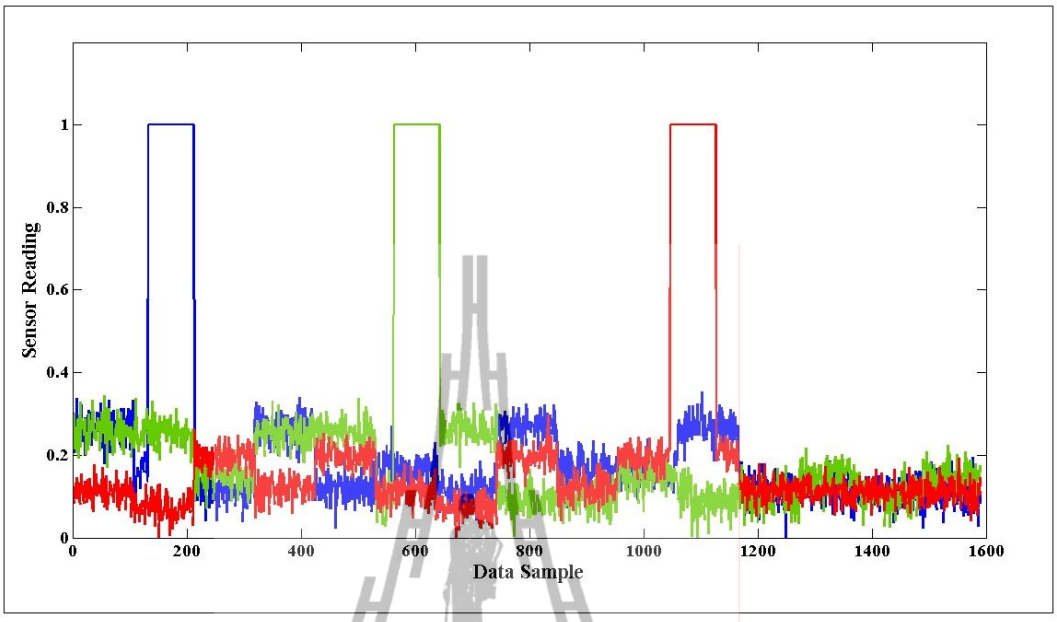


Figure 4.10 3KPI Synthetic data with 80/1 fault.

4.4.1.2 Real world datasets

For the real world datasets, we did not have ground truth information about the actual faults in each dataset. Therefore, we used the histogram method to separate the normal data from abnormal data**.**

1) INTEL dataset: 54 Mica2Dot motes with temperature, humidity light and voltage were deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004 (The Intel Lab, Online, 2004). In the experiment, we selected 20,000 samples from temperature, humidity and voltage readings. In the first part of 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. In the second part, we presented the humidity and voltage readings. We selected the threshold value of 24 and 47 as the

upper and lower bounds for humidity readings, and 2.5 and 2.8 for voltage readings. Using the histogram method, we found that this dataset had short faults.

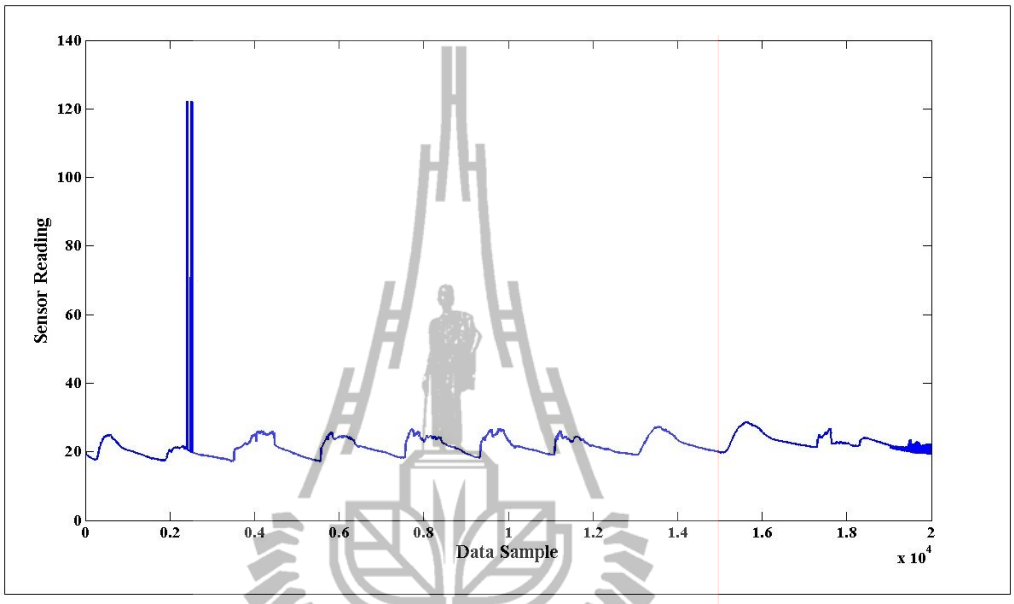


Figure 4.11 INTEL dataset (temperature reading).

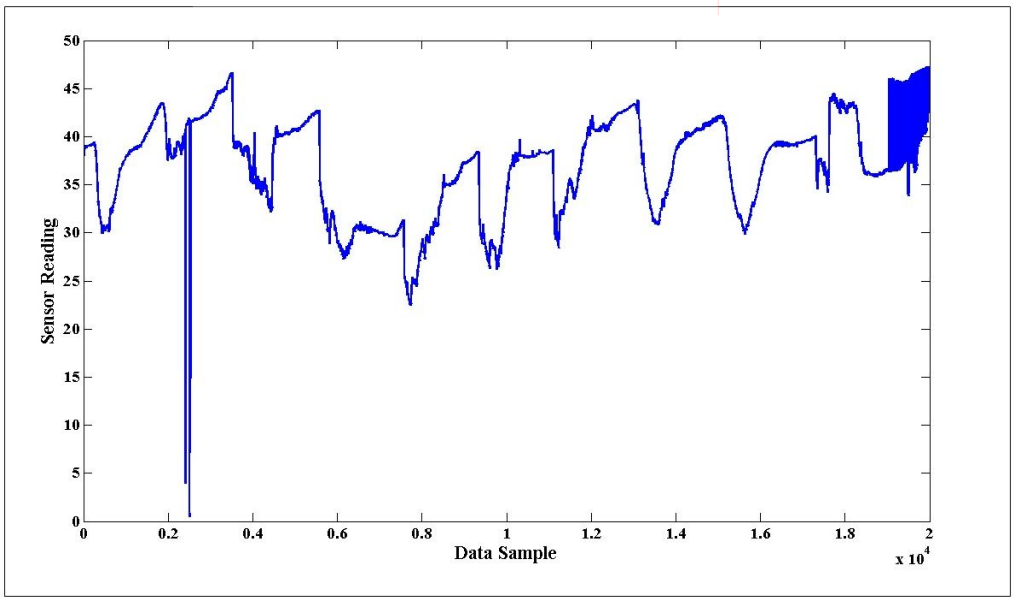


Figure 4.12 INTEL dataset (humidity reading).

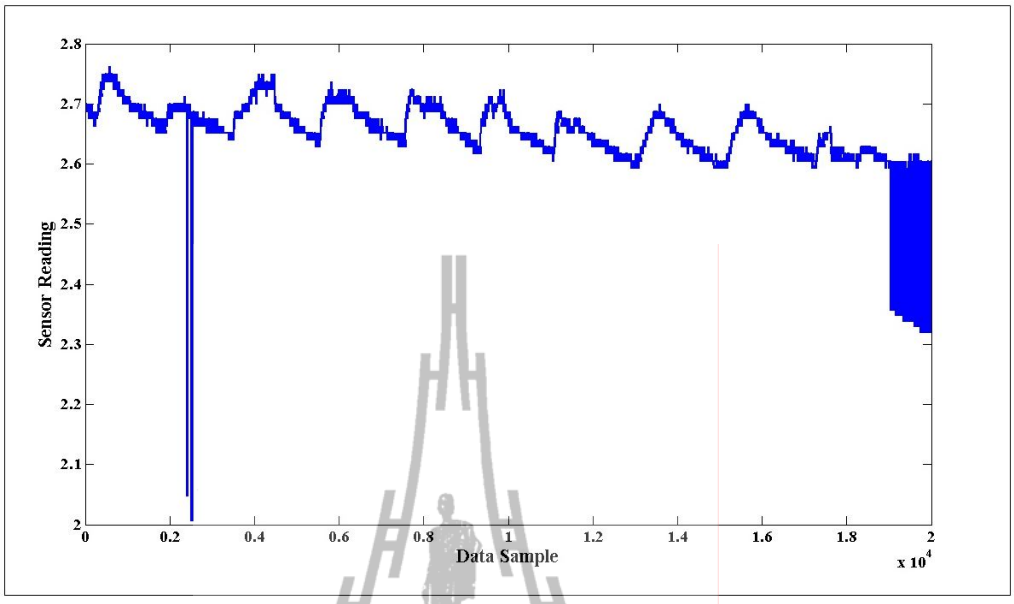


Figure 4.13 INTEL dataset (voltage reading).

2) NAMOS dataset: In this dataset, 9 buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, for over 24 hours in August, 2006 (Network Aquatic Microbial Observing System, Online, 2006). We analyzed 10,000 sample measurements from fluorimeters on buoys no. 103. In the first part of experiment, we used the fluorimeter readings and selected the threshold of 0 and 500 as lower and upper bounds of the normal region, respectively. By considering the positions of anomalous data, we found that constant faults were present in this dataset. In the second part, we selected the readings from 2 temperature sensors and considered them as normal because the histogram method cannot clearly separate normal and anomalous region.

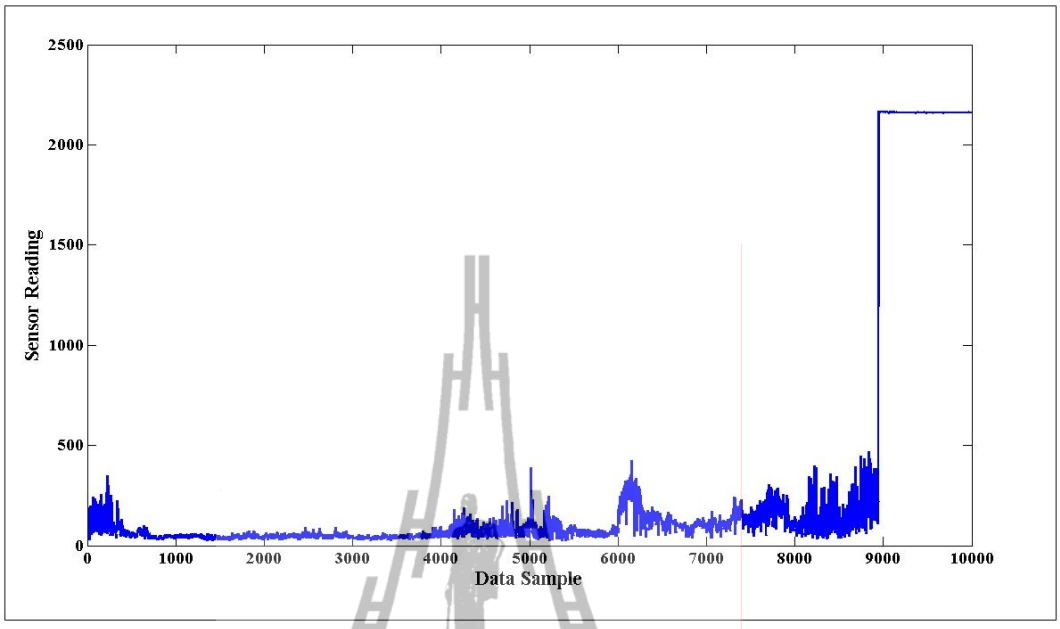


Figure 4.14 NAMOS dataset (fluorimeter reading).

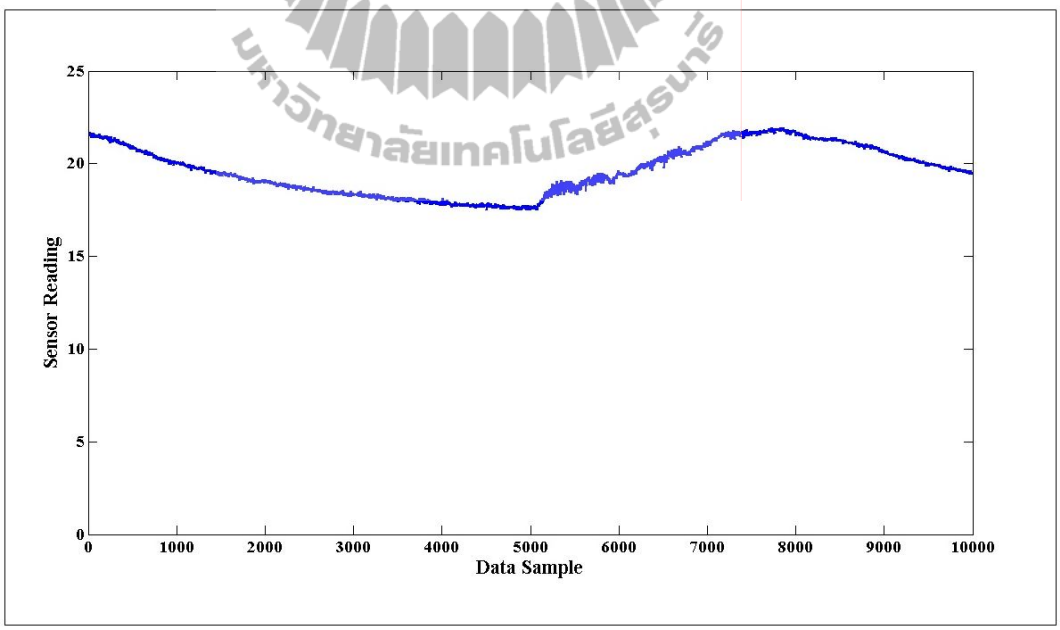


Figure 4.15 NAMOS dataset (temperature reading from sensor 1).

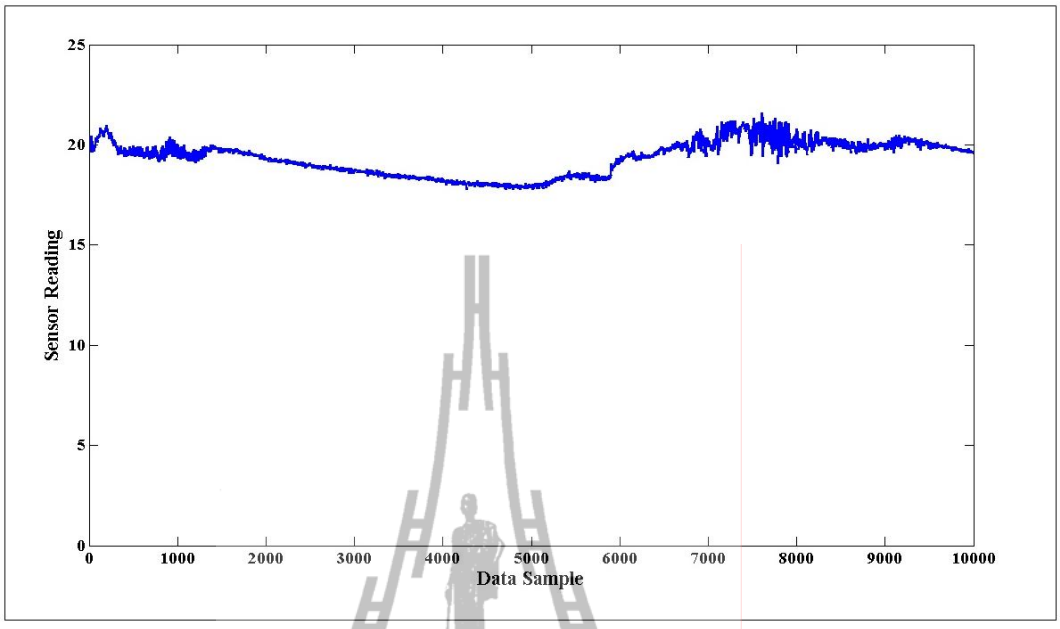


Figure 4.16 NAMOS dataset (temperature reading from sensor 2).

3) SensorScope pdg2008-metro-1 (pdg2008) dataset: In the first part of experiment, we used two types (KPIs) of data in the pdg2008 dataset for anomaly detection, i.e., the surface and ambient temperature readings. The lower and upper threshold values used for anomaly detection were -12 and 4 for the ambient temperature and -14 and 4 for the surface temperature. By considering the positions of anomalous data, we found that this dataset contained noise faults. In the second part, we included the solar radiation readings as normal data, since the histogram method cannot clearly separate normal and anomalous region.

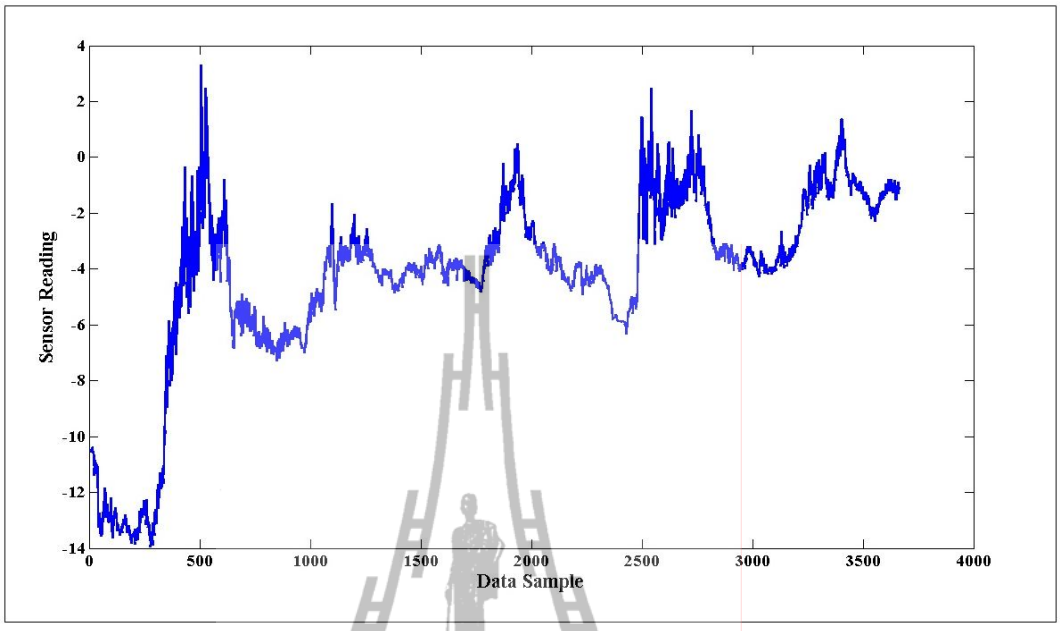


Figure 4.17 pdg2008 dataset (ambient temperature reading).

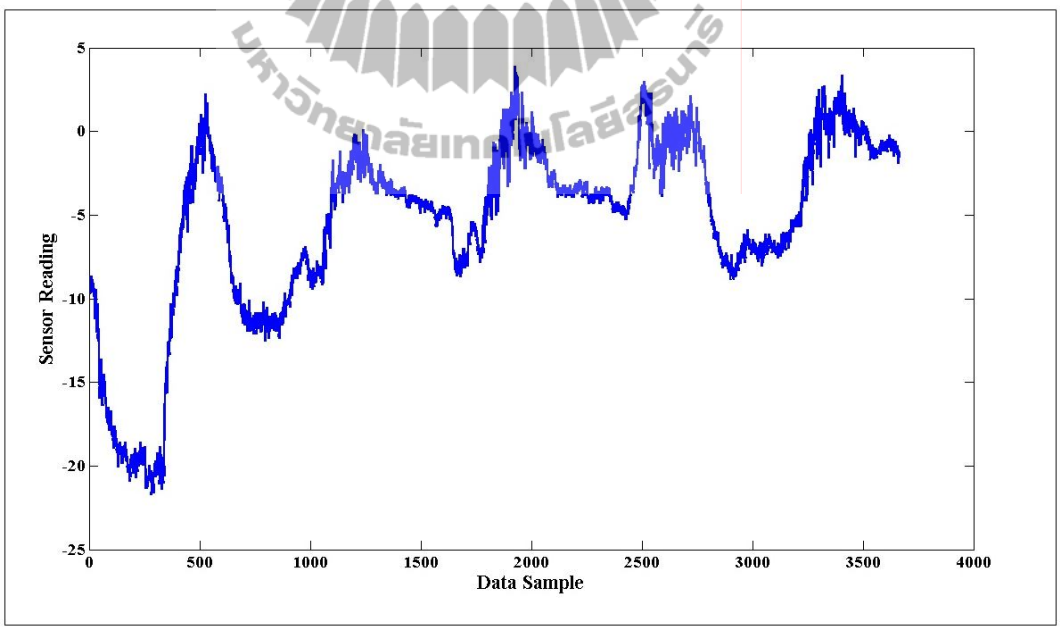


Figure 4.18 pdg2008 dataset (surface temperature reading).

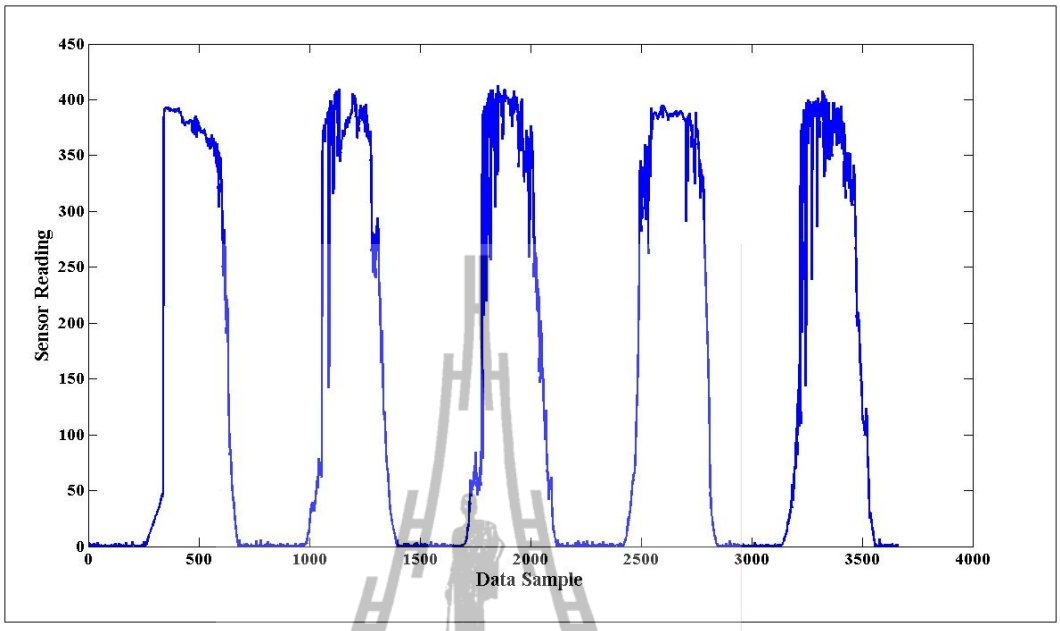


Figure 4.19 pdg2008 dataset (solar radiation reading).

4.4.2 Performance evaluation

This research proposed the integration between OCSVM anomaly detection and data compression using LWT, and then compared it with the OCSVM + DWT and OCSVM + PCA. For the OCSVM + LWT and OCSVM + DWT. We replaced the original set of input data with low pass or high pass LWT coefficients by using Haar mother wavelet. The low (high) pass wavelet coefficients obtained were referred to as “low (high) pass data” with the same number of KPIs and with a length (i.e., observation length) of just half of the original data vector. The original data were referred to as “uncompressed data”.

For the OCSVM with PCA algorithm, we replaced the original set of input data with the PCA 1KPI (2KPI, 3KPI) dataset which had 1 KPI (2 KPIs, 3 KPIs) with the same observation length as the original data vector. We referred them respectfully as *PCA 1KPI (2KPI, 3KPI) data*.

Figure 4.20 shows the 2KPI synthetic dataset injected with short faults results. Figure 4.21 shows the results for the 1KPI INTEL dataset containing short faults, which agreed with results for synthetic data injected with 1/80 short faults, where all algorithms performed equally well. This was because short faults had high amplitude which can be easily detected.

Figure 4.22 is the result for the synthetic dataset injected with noise faults. Results showed that all algorithms performed equally well except for OCSVM+DWT (HP) and OCSVM+LWT (HP) which gave the worst performance.

Figure 4.23 shows results for the pdg2008 dataset. Since faults in the pdg2008 dataset had lower amplitude which were more difficult to detect, OCSVM alone and OCSVM+PCA did not perform as well as OCSVM+DWT (LP) and OCSVM+LWT (LP). On the other hand, OCSVM+DWT (HP) and OCSVM+LWT (HP) gave the worst performance which agreed with synthetic data results. This was because HP coefficients reflect the rate of changes between two successive samples. Therefore, HP coefficients were more suitable for short faults whereas LP coefficients were more suitable for slower changing faults like noise faults.

Figures 4.24 and 4.25 show the synthetic dataset injected with constant faults and the NAMOS dataset results, respectively. The results for NAMOS dataset agreed with the results for synthetic dataset injected with constant faults. The OCSVM+LWT [LP] and OCSVM+DWT [LP] performed better than the OCSVM alone and OCSVM+PCA. The OCSVM+LWT [HP] and OCSVM+DWT [HP] gave the worst performance. Note that the results for constant faults were similar to the results for noise faults. This was because both types of faults were trend-like changes

and therefore their features were more significant when captured with LP coefficients than HP coefficients.

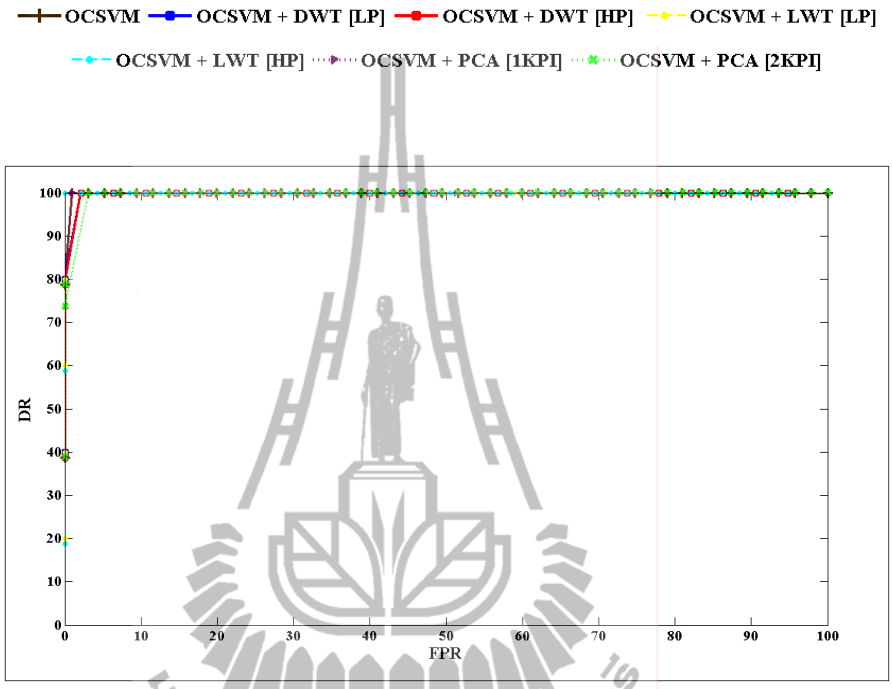


Figure 4.20 ROC curve for 2KPI Synthetic data injected with 1/80 faults.

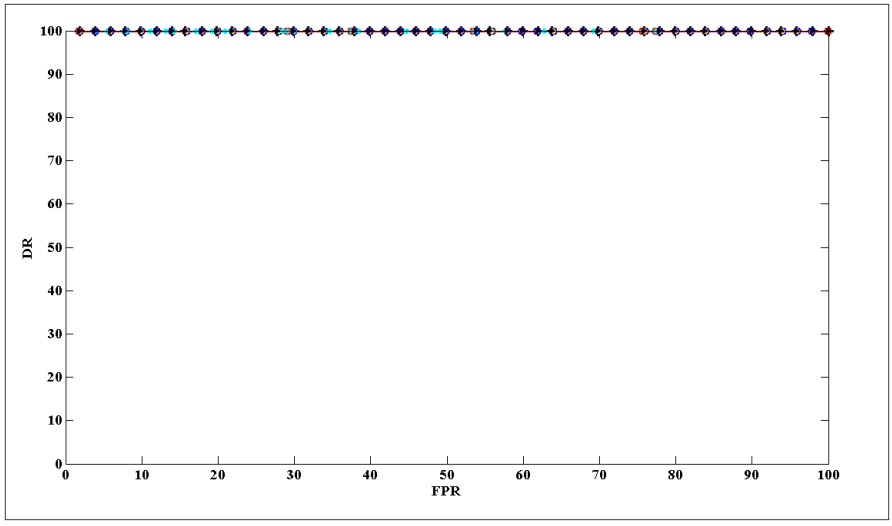


Figure 4.21 ROC curve for 1KPI INTEL dataset (containing short faults).

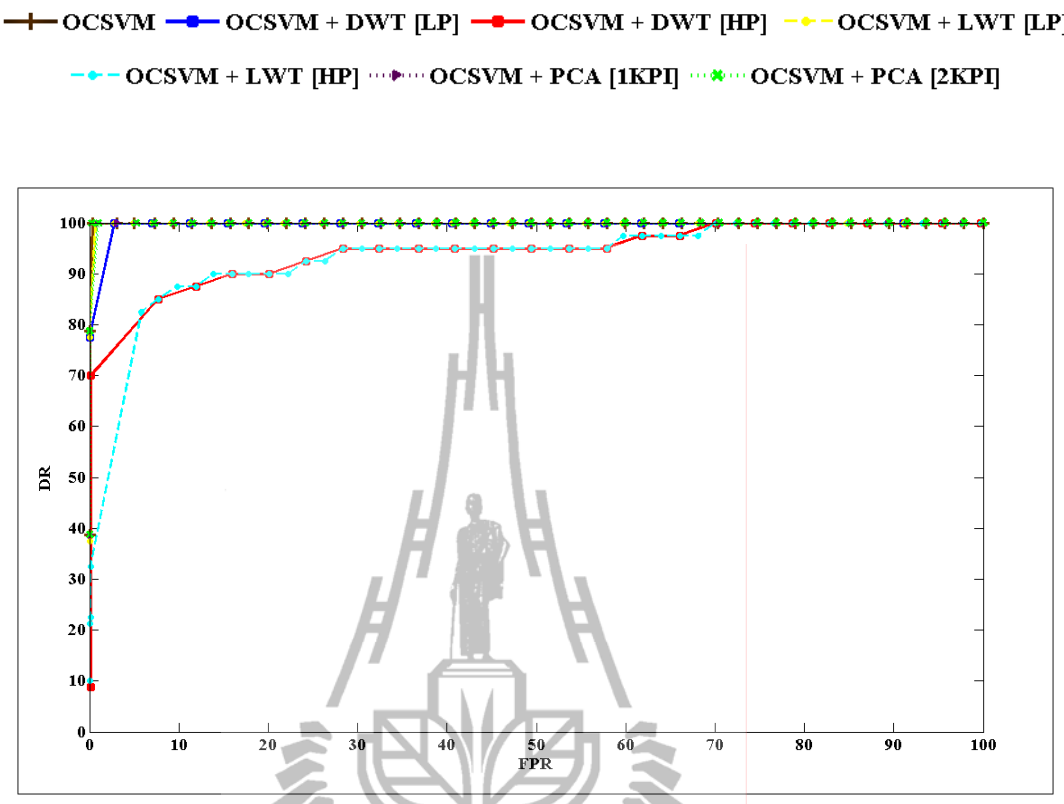


Figure 4.22 ROC curve for 2KPI Synthetic data injected with 20/4 faults.

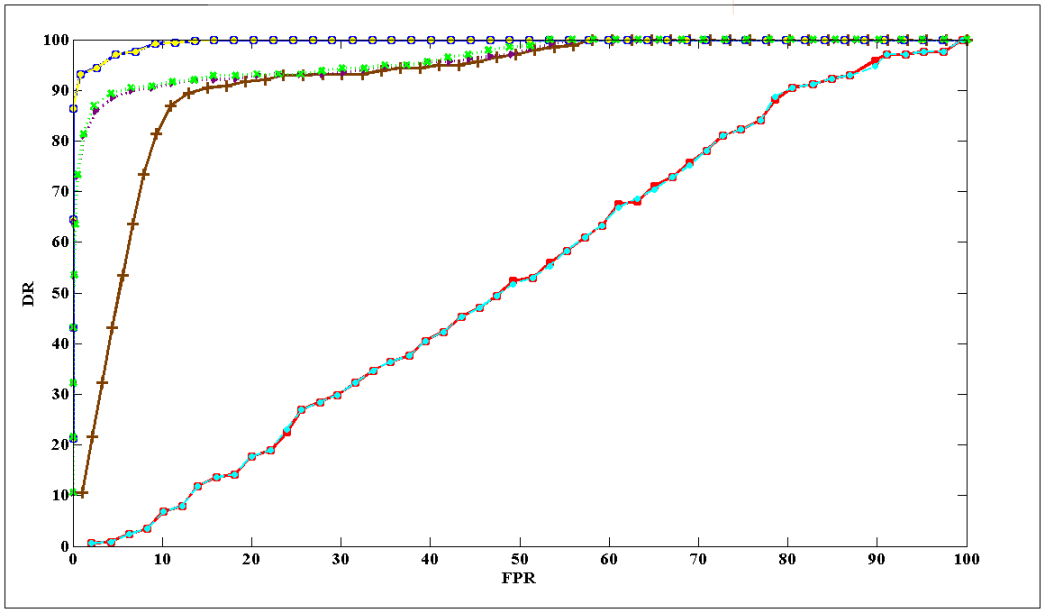


Figure 4.23 ROC curve for 2KPI pdg2008 dataset (containing noise faults).

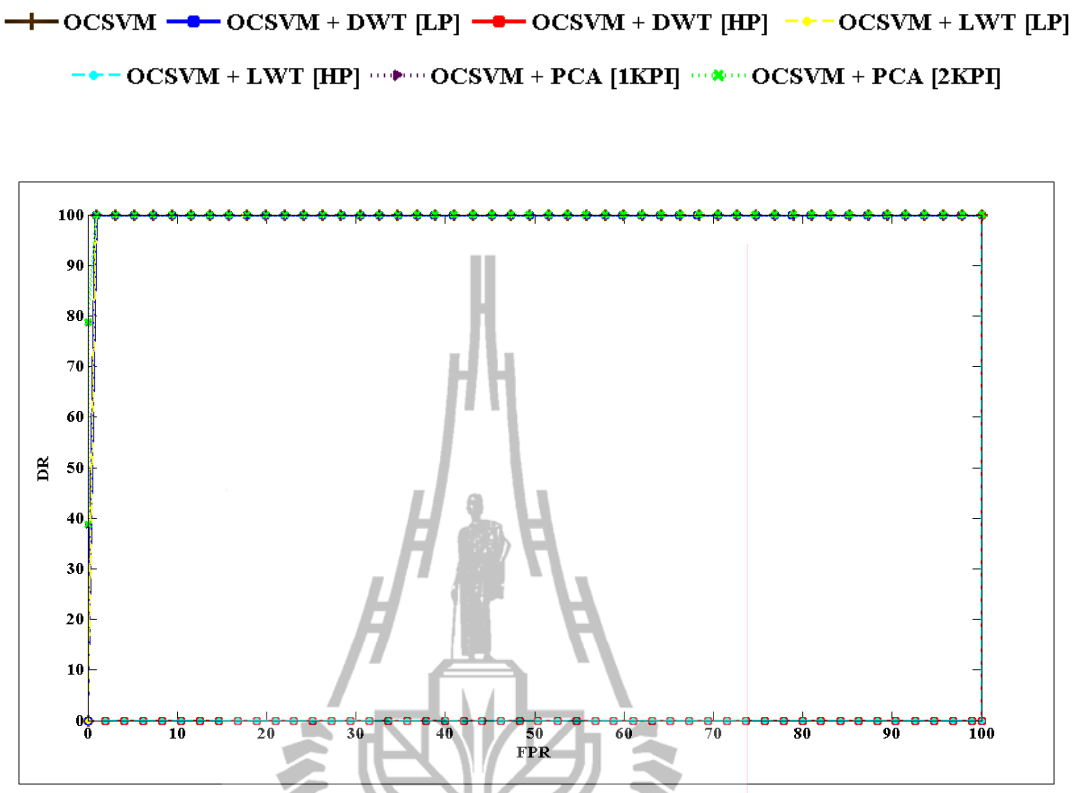


Figure 4.24 ROC curve for 2KPI Synthetic data injected with 80/1 fault.

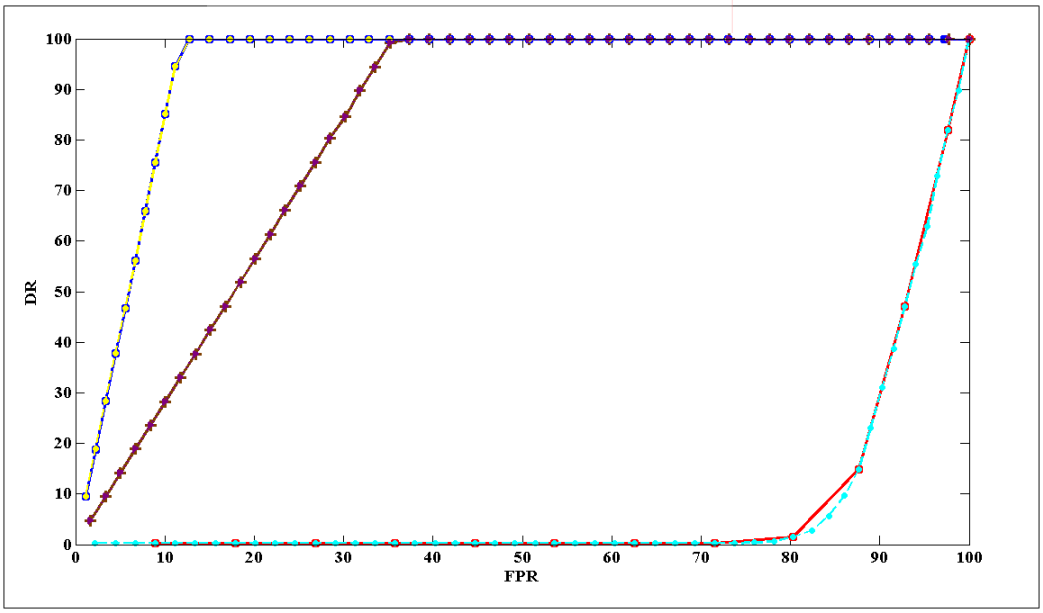


Figure 4.25 ROC curve for 1KPI NAMOS dataset (containing constant fault).

4.4.3 Extending to 3 KPI datasets

Figure 4.26 shows the 3KPI synthetic data injected with short faults results which all algorithms perform equally well. This is because the short fault has high amplitude which can be easily detected, except for the OCSVM+PCA [1 KPI] which gave the worst performance. This is because the synthetic data was generated by different injected fault positions, the appearance of data in each KPI were different (see Figure 4.6). Compression to PCA with just 1KPI ignored some components of the data, and the appearance of data was thus distorted. Therefore, the positions of faults were distorted making it difficult to detect.

Figure 4.27 shows the 3KPI INTEL dataset which contained short faults in 3KPIs results. The results showed that all algorithms performed equally well. This is because HP coefficients reflect the rate of changes between two successive samples and the short fault has high amplitude. Therefore, the short fault can be easily detected. Furthermore, each KPI of INTEL dataset contained faults in the same position (see Figure 4.11-4.13). Compression to PCA with 1KPI ignored some components of the data thereby distorting its appearance. However, the positions of the faults still clearly appear and can be easily detected although compressed to PCA 1KPI.

Figure 4.28 shows the 3KPI synthetic data injected with noise faults results. The results shows that the OCSVM+LWT [LP] and OCSVM+DWT [LP] performed equally well as the OCSVM alone, OCSVM+PCA [2KPI and 3KPI]. The OCSVM+LWT [HP] and OCSVM+DWT [HP] gave worse performance than these algorithm. This is because HP coefficients reflect the rate of changes between two successive samples. Therefore, HP coefficients were more suitable for short faults

whereas LP coefficients were more suitable for slower changing faults like noise faults. In addition, The OCSVM+PCA [1 KPI] gave the worst performance of all. Since, the synthetic data was generated by different injected fault positions, the appearance of data in each KPI are different (see Figure 4.8). Compression to PCA [1KPI] distorted the data. Therefore, the positions of faults were distorted making it difficult to detect.

Figure 4.29 shows the results for the 3 KPI pdg2008 dataset which contained noise faults. The OCSVM+LWT [LP] and OCSVM+DWT [LP] performed better than the OCSVM alone and OCSVM+PCA. The OCSVM+LWT [HP] and OCSVM+DWT [HP] gave the worst performance while the synthetic data result in Figure 4.28 show that the OCSVM+PCA [1KPI] gave the worst performance. This was because HP coefficients reflect the rate of changes between two successive samples. Moreover, the pdg2008 dataset (see Figure 4.17-4.18) had lower noise amplitude than the synthetic data (see Figure 4.8). Therefore, the 3KPI pdg2008 dataset results shows that the OCSVM+LWT [HP] and OCSVM+DWT [HP] gave the worst performance.

Figures 4.30 show the 3KPI synthetic dataset injected with constant faults result. All algorithm perform equally well except for the OCSVM+LWT [HP] and OCSVM+DWT [HP] which gave the worst performance.

Figures 4.31 show the 3KPI NAMOS dataset results which agreed with results for synthetic data in Figure 4.30. The OCSVM+LWT [LP] and OCSVM+DWT [LP] performed better than the OCSVM alone and OCSVM+PCA. The OCSVM+LWT [HP] and OCSVM+DWT [HP] gave the worst performance. Note that the results for constant faults were similar to the results for noise faults. This was because both types

of faults were trend-like changes and therefore their features were more significant when captured with LP coefficients than HP coefficients.

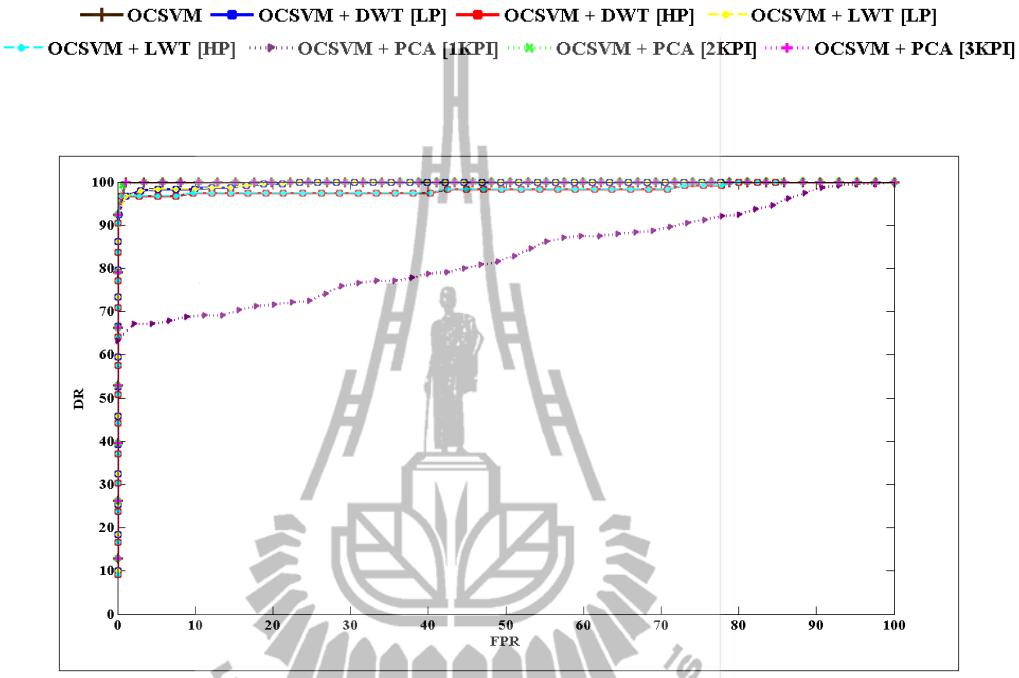


Figure 4.26 ROC curve for 3KPI Synthetic data injected with 1/80 faults.

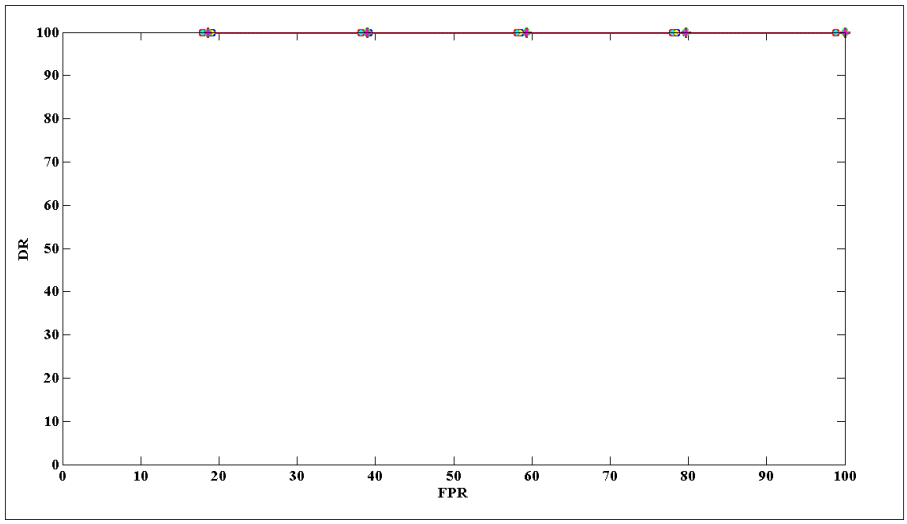


Figure 4.27 ROC curve for 3KPI INTEL dataset (containing short fault).

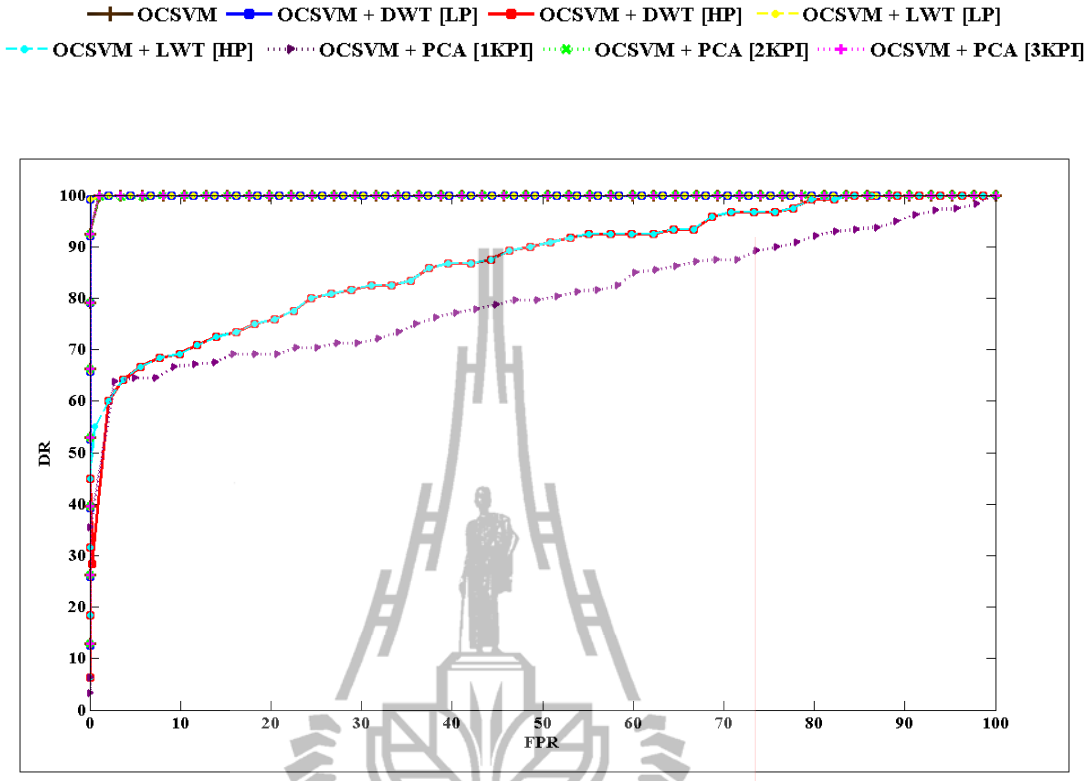


Figure 4.28 ROC curve for 3KPI Synthetic data injected with 20/4 faults.

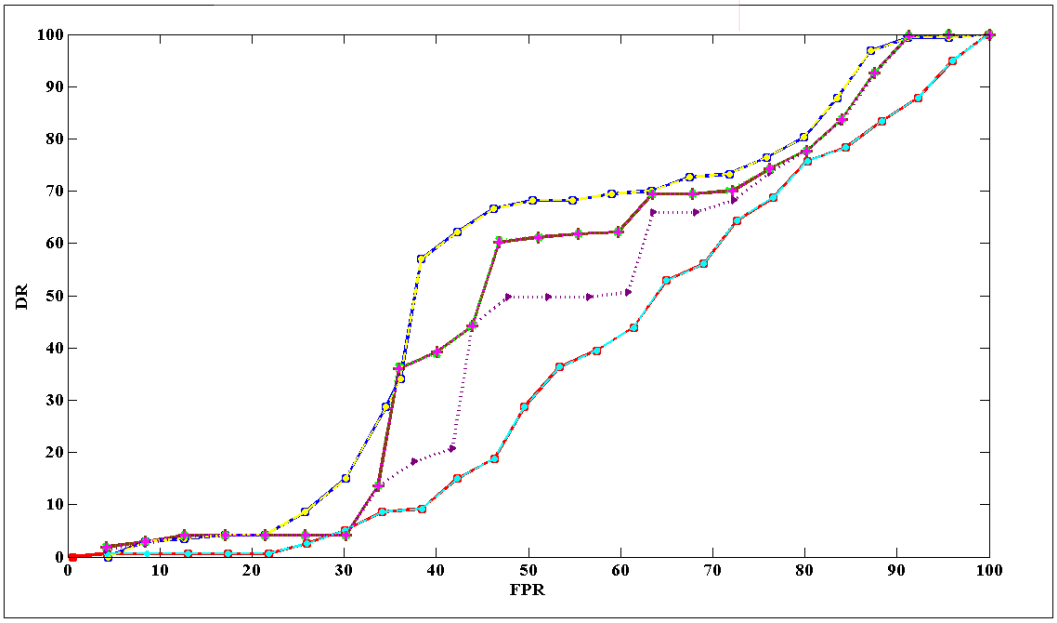


Figure 4.29 ROC curve for 3KPI pdg2008 dataset (containing noise fault).

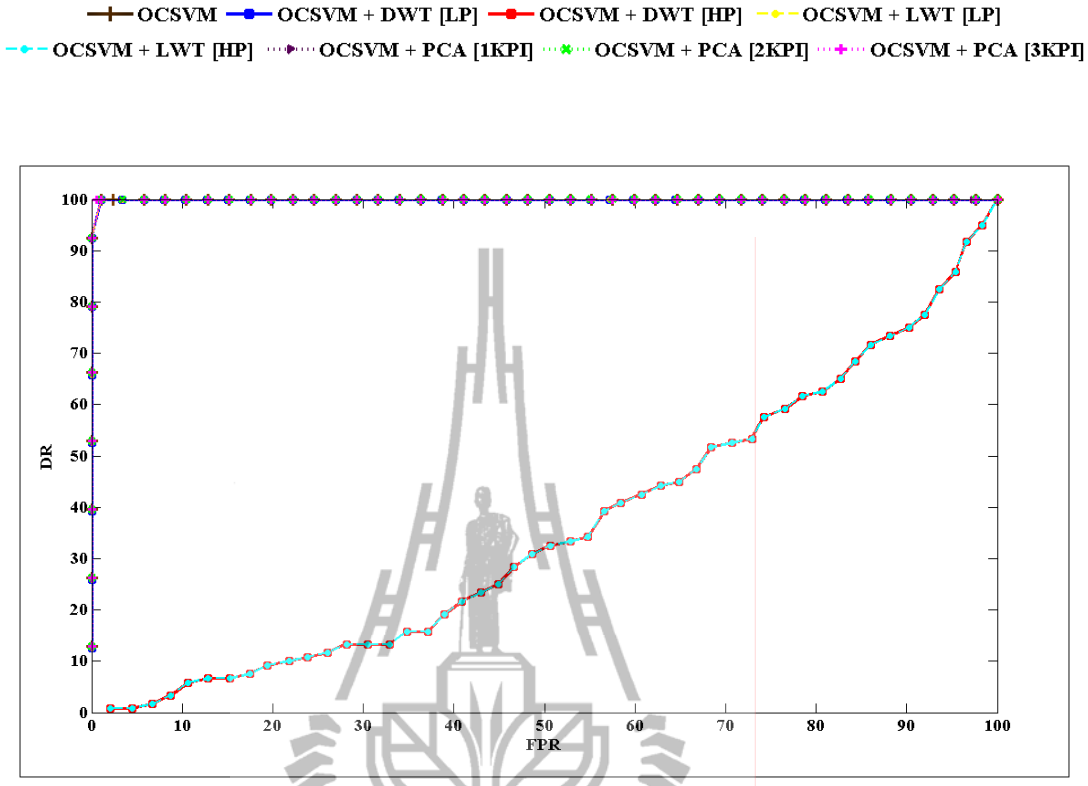


Figure 4.30 ROC curve for 3KPI Synthetic data injected with 80/1 fault.

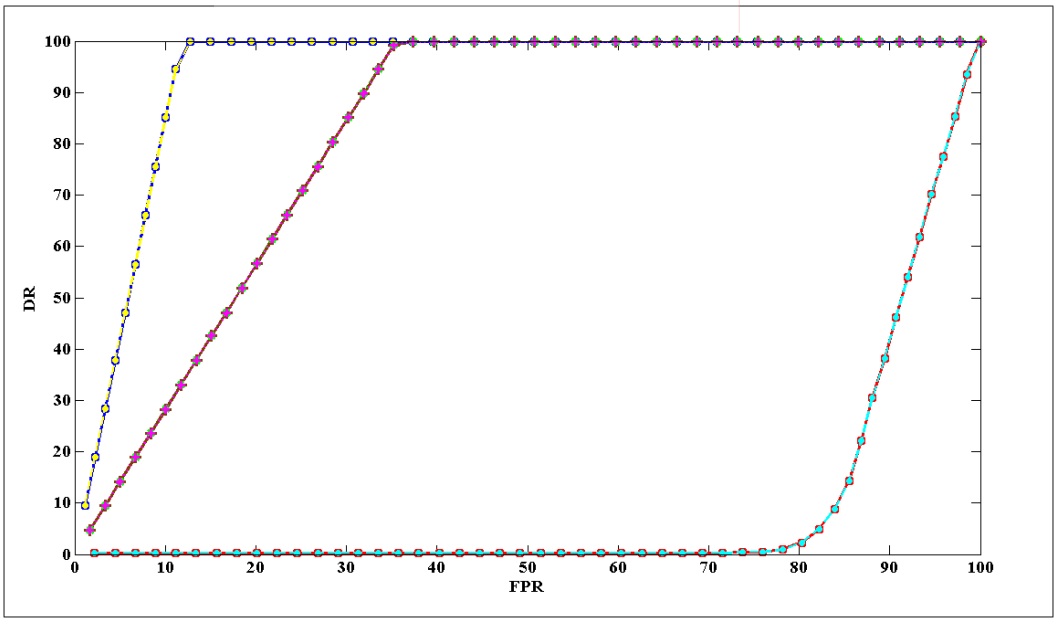


Figure 4.31 ROC curve for 3KPI NAMOS dataset (containing constant fault).

4.4.4 Computation time evaluation

There are many researchers have mentioned that the data compression technique using DWT which has been implemented by convolutions, require a larger number of arithmetic computation, storage space and more computation time than LWT (Sweldens, 1998; Achaya and Chakrabarti, 2006; X. L. Li, J. W. Zhang, and W. H. FANG, 2009). Therefore, an experiment was conducted by repeatedly feeding runs of vectors of 1590 data elements into each data compression method in order to measure the computational time required for each method.

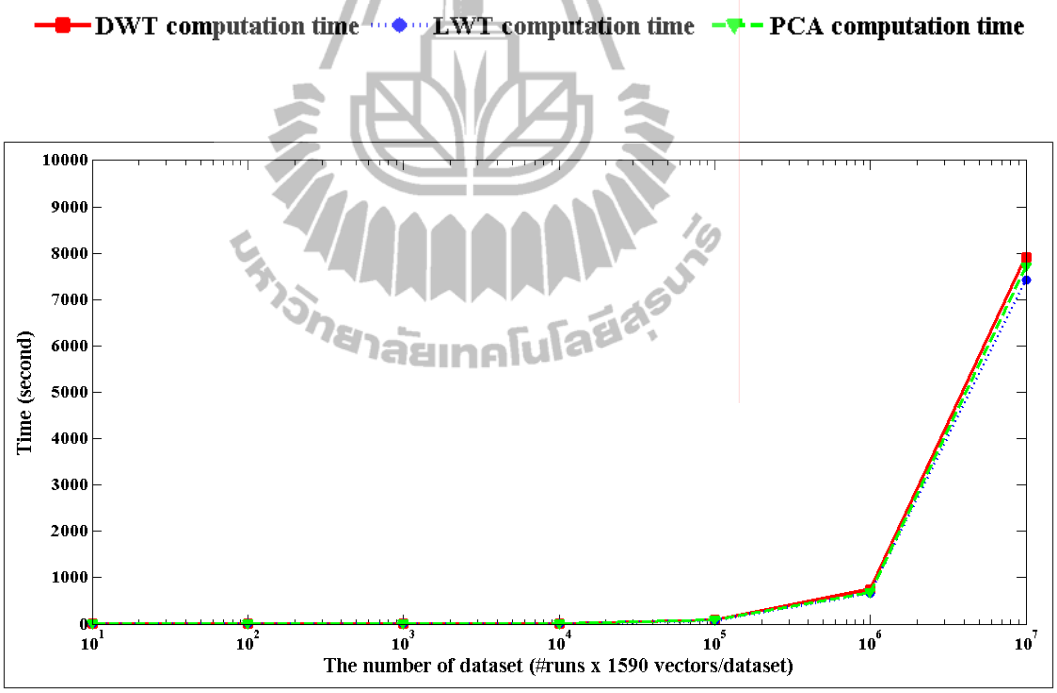


Figure 4.32 Computation time of each data compression technique

Figure 4.32 shows the computation time each technique spends for processing each magnitude increase of runs of the input dataset. Note that LWT used less computation time than DWT and PCA.

4.5 Conclusion

We proposed the integration of OCSVM with LWT for anomaly detection in WSNs. We numerically evaluated the algorithm using MATLAB and tested it with both synthetic data and real world datasets. For synthetic data and real world dataset with the short faults, the OCSVM with LWT performed equally well as OCSVM alone, OCSVM with DWT and OCSVM with PCA. For synthetic data and real world dataset with the noise and constant faults, the OCSVM with LWT [LP] and the OCSVM with DWT [LP] gave better performance than the OCSVM alone and OCSVM with PCA, while the OCSVM with LWT [HP] and the OCSVM with DWT [HP] gave the worst performance.

However, LWT had a simpler computation and required less computation time than DWT. Therefore, OCSVM with LWT is more suitable for WSN.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This thesis proposed an integration of an anomaly detection technique called the one-class support vector machines (OCSVM) and a data compression techniques called the discrete wavelet transform (DWT) and lifting wavelet transform (LWT), to compress data and detect anomalies in wireless sensor networks (WSNs). Our proposed algorithm was designed for sending the compressed anomaly measurement data to the base station in order to help reduce wasted energy caused by transmitting all the measurement data. We numerically evaluated the algorithm using MATLAB and tested it with both synthetic data and real world datasets which contained three types of real-world fault including short, noise and constant faults. Our experiment was divided into 2 parts which can be summarized as follows.

5.1.1 Anomaly detection DWT coefficients

The OCSVM was integrated with DWT first in order to study the effect of data compression on anomaly detection. We found that this integration can increase the efficiency of anomaly detection by achieving accurate detection rate (DR) while transmitting just half of the original data length. For synthetic data, the OCSVM+DWT with LP coefficients achieved 100% DR with marginal increase in false positive rate (FPR) when compared with all other algorithms which including the OCSVM alone, OCSVM+DWT with HP coefficients and the self-organizing map

(SOM)-based algorithm. For real world datasets, the OCSVM+DWT with LP coefficients performed best by achieving nearly 100% DR although with slightly higher FPR for datasets containing short and noise faults. These results suggested that with data compression and using just half of the data input, OCSVM+DWT (LP) algorithm was suited for short and noise faults whereas SOM+DWT (LP) was suited for short and constant faults.

5.1.2 Anomaly detection LWT coefficients

From the study, we studied a second generation wavelet transform data compression technique called the lifting wavelet transform (LWT) which required simpler computation, less memory space and lower computation time than DWT. We integrated the OCSVM with LWT and tested it with both synthetic data and real world datasets. For synthetic data and real world dataset with short faults, the OCSVM with LWT performed equally well as OCSVM alone, OCSVM with DWT and OCSVM with PCA. For synthetic data and real world dataset with noise and constant faults, the OCSVM with LWT [LP] and the OCSVM with DWT [LP] gave better performance than the OCSVM alone and OCSVM with PCA. On the contrary the OCSVM with LWT [HP] and the OCSVM with DWT [HP] gave the worst performance. It was also demonstrated that LWT was less demanding in term of computation and computation time than DWT. Our results therefore suggested that OCSVM with LWT was more suitable for implementation in WSNs.

5.2 Future work

In the future, there are issues worthwhile investigating.

5.2.1 Increasing DWT and LWT level

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

5.2.2 Exploring other types of wavelets

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

5.2.3 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.

5.2.4 Comparison with other data compression techniques

WSNs are resource constrained, i.e., with limited power supply, bandwidth for communication, processing speed, and memory space. One possible way of achieve maximum utilization of such resources is to apply data compression on sensor data (Kimura, and Latifi, 2005; Sadler, and Martonosi 2006). Alternative data compression algorithms for anomaly detection in WSNs in the recent literature may be worthwhile investigting.

5.2.5 Enhancing to fault predictability

The anomaly detection algorithm in this thesis can support detection when faults have already occurred. A challenging issue is not only to be able to detect

faults when they have already occurred but to predict them before a fault actually occurs or fault prediction. Such extension allows the user to take a suitable course of action to prevent the monitored environment before any significant damage occurs.

5.2.6 Normalize data prior feeding to anomaly detection

From the experiments carried out in this thesis, it was found that data preprocessing by means of normalization can improve the performance of certain anomaly detection algorithms. Therefore, a worthwhile future direction may also include studying alternative normalization and data preprocessing method prior to anomaly detection.

REFERENCES

Acharya T. and Chakrabarti C. (2006). A Survey on Lifting-Based Discrete Wavelet Transform Architectures. In **Proceedings of the Journal of VLSI Signal Processing.** 42: 321-339.

Arici, T., Gedik, B., Altunbasak, Y., and Liu, L. (2003). PINCO: a Pipelined In-Network Compression Scheme for Data Collection in Wireless Sensor Networks. In **Proceedings of the 12th International Conference on Computer Communications and Networks.** (pp. 539-544).

Boukerche, A., and Samarah, S. (2009). In-Network Data Reduction and Coverage-Based Mechanisms for Generating Association Rules in Wireless Sensor Networks. In **Proceedings of the IEEE Transactions on Vehicular Technology.** 58 (8): 4426-4438.

Bruce, L. M., Koger, C. H., and Li, J. (2002). Dimensionality Reduction of Hyperspectral Data Using Discrete Wavelet Transform Feature Extraction. In **Proceedings of the IEEE Transactions International Geosciences and Remote Sensing.** 40 (10): 2331-2338.

Cai, W., and Zhang, M. (2008). Data Aggregation Mechanism based on Wavelet-entropy for Wireless Sensor Networks. In **Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing.** (pp. 1-4).

Capo-Chichi, E. P., Guyennet, H., and Friedt, J. M. (2009). K-RLE : A new Data Compression Algorithm for Wireless Sensor Network. In **Proceedings of the Third International Conference on Sensor Technologies and Applications.** (pp. 502-507).

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly Detection: A Survey. In **Proceedings of the ACM Computing Surveys.** 41 (3): 1 - 72.

Ciancio, A., and Ortega, A. (2004). A distributed wavelet compression algorithm for wireless sensor networks using lifting. In **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2004.** 4: IV-633-IV-636.

Ciancio, A., and Ortega, A. (2005). A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. In **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2005.** 4: IV-825-IV-828.

Doshi, R. A., King, R. L., and Lawrence, G. W. (2007). Wavelet-SOM in Feature Extraction of Hyperspectral Data for Classification of Nematode Species. In **Proceedings of the IEEE International Conference on Geosciences and Remote Sensing Symposium.** (pp. 2818-2821).

Du, P., Tan, K., and Xing, X. (2010). Wavelet SVM in Reproducing Kernel Hilbert Space for hyper spectral remote sensing image classification. In **Proceedings of the Optics Communications.** 283 (24): 4978-4984.

Dwinnell , W. (2010). **Principal Components Analysis.** [Online]. Available: http://matlabdatamining.blogspot.com/2010/02/principal-components-analysis.html

Goh, H.G., Sim, M.L., and Ewe, H.T. (2007). Agriculture monitoring. In N. P. Mahalik (ed.). **Sensor Networks and Configuration** (pp. 439-462). New York: Springer.

Hsu, C. W., Chang, C. C., and Lin, C. J. (2003). **A Practical Guide to Support Vector Classification.** [Online]. Available: http://www.csie.ntu. edu.tw/~cjlin

Kimura, N., and Latifi, S. (2005). A Survey on Data Compression in Wireless Sensor Networks. In **Proceedings of the International Conference on Information Technology: Coding and Computing.** 2: 1-6.

Kiziloren, T., and Germen, E. (2009). Anomaly detection with Self-Organizing Maps and effects of Principal Component Analysis on feature vectors. In **Proceedings of the 5th International Conference on Natural Computation.** 6: 509-513.

Laskov, P., Schafer, C., and Kotenko, I. (2004). Intrusion detection in unlabeled data with quarter sphere support vector machines. In **Proceedings of the Detection of Intrusions and Malware & Vulnerability Assessment.** 27 (4): 228-236.

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. In **Proceedings of the International Conference on Apperceiving Computing and Intelligence Analysis.** (pp. 228-233).

Lin, S., Gunopulos, D., Kalogeraki, V., and Lonardi, S. (2005). A Data Compression Technique for Sensor Networks with Dynamic Bandwidth Allocation. In **Proceedings of the 12th International Symposium on Temporal Representation and Reasoning.** (pp. 186-188).

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.

Lutsa, J., et al. (2010). A tutorial on support vector machine-based methods for classification problems in chemometrics. In **Proceedings of the Analytica Chimica Acta.** 665 (2): 129-145.

Maleki, A., and Jalali, S. (2005). **Directional lifting-based wavelet transforms.** [Online]. Available: http://scien.stanford.edu/pages/labsite/2005/ee398/projects/reports/Jalali%20Maleki%20-%20Project%20Report%20-%20DIRECTIONAL%20LIFTING-BASED%20WAVELET.PDF

Manjunath, A., and Ravikumar, H. M. (2010). Comparison of discrete wavelet transform (DWT) lifting wavelet transform (LWT) stationary wavelet transform (SWT) and S-Transform in power quality analysis. In **Proceedings of the European Journal of Scientific Research.** 39 (4): 569-576.

Marcelloni, F., and Vecchio, M. (2008). A Simple Algorithm for Data Compression in Wireless Sensor Networks. In **Proceedings of the IEEE Communications Letters.** 12 (6): 411-413.

Min, L., and Dongliang, W. (2009). Anomaly Intrusion Detection Based on SOM. In **Proceedings of the WASE International Conference on Information Engineering.** (pp. 40-43).

Rajasegarar, S., Leckie, C., Palaniswami, M., and Bezdek, J. C. (2006). Distributed Anomaly Detection in Wireless Sensor Networks. In **Proceedings of the 10th IEEE Singapore International Conference on Communication systems.** (pp. 1-5).

Rajasegarar, S., Leckie, C., Palaniswami, M., and Bezdek, J. C. (2007). Quarter sphere based distributed anomaly detection in wireless sensor networks. In **Proceedings of the IEEE International Conference on Communications.** (pp. 3864-3869). New Jersey, USA: IEEE Operations Center.

Rajasegarar, S., Leckie, C., and Palaniswami, M. (2008). Anomaly detection in wireless sensor networks. In **Proceedings of the IEEE Wireless Communications**. 15 (4): 34-40.

Rajasegarar, S., Leckie, C., Palaniswami, M., and Bezdek, J. C. (2010). Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines for Anomaly Detection in Sensor Network. In **Proceedings of the IEEE Transactions on Information Forensics and Security.** 5 (3): 518-533.

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.

Sharma, A.B., Golubchik, L., and Govindan, R. (2010). Sensor faults: detection methods and prevalence in real-world datasets. In **Proceedings of the ACM Transactions on Sensor Networks.** 6 (3): 1-34.

Siripanadorn, S., Hattagam, W., and Teaumroong, N. (2010) Anomaly detection in wireless sensor networks using Self-Organizing Map and Wavelets. In **Proceedings of the International Journal of Communications**. 4 (3): 74-83.

Smith, L. I. (2002). **A tutorial on Principal Components Analysis.** [Online]. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Sweldens, W. (1998). The lifting scheme: a construction of second generation wavelets. In **Proceedings of the Siam Journal on Mathematical Analysis.** 29: 511-546.

Tax, D. M. J., and Duin, R. P. W. (2004). Support vector data description. . In **Proceedings of the Machine Learning.** 54 (1): 45-66.

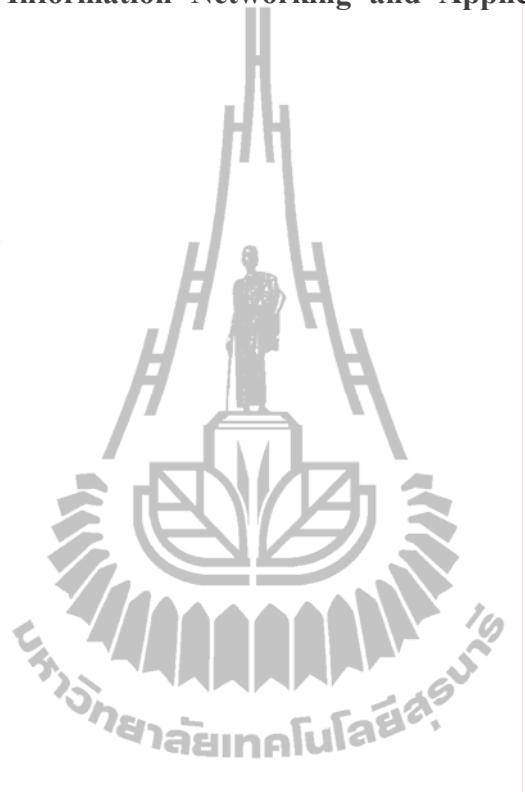
Wang, F., Qian, Y., Dai, Y., and Wang, Z. (2010). A Model Based on Hybrid Support Vector Machine and Self-Organizing Map for Anomaly Detection. In **Proceedings of the 2010 International Conference on Communications and Mobile Computing.** 1: 97-101.

Watson, M. J., Liakopoulos, A., Brzakovic, D., and Georgakis, C. (1995). Wavelet techniques in the compression of process data. In **Proceedings of the American Control Conference.** 2: 1265-1269.

Xu, Y., and Chow, T. W.S. (2010). Efficient Self-Organizing Map Learning Scheme Using Data Reduction Preprocessing. In **Proceedings of the World Congress on Engineering 2010.** 2183 (1): 273-276.

Yao, Y., Sharma, A.B., Golubchik, L., and Govindan, R. (2010). Online Anomaly Detection for Sensor Systems: a Simple and Efficient Approach. In **Proceedings of the Journal Performance Evaluation.** 67 (11): 1-24.

Zhang, Y., Meratnia, N., and Havinga, P. (2009). Adaptive and Online One-Class Support Vector Machine-based Outlier Detection Techniques for Wireless Sensor Networks. In **Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops.** (pp. 990-995).



APPENDIX A

DATASETS FOR EXPERIMENT CHAPTER 3

Datasets for Chapter 3

1. Synthetic Data

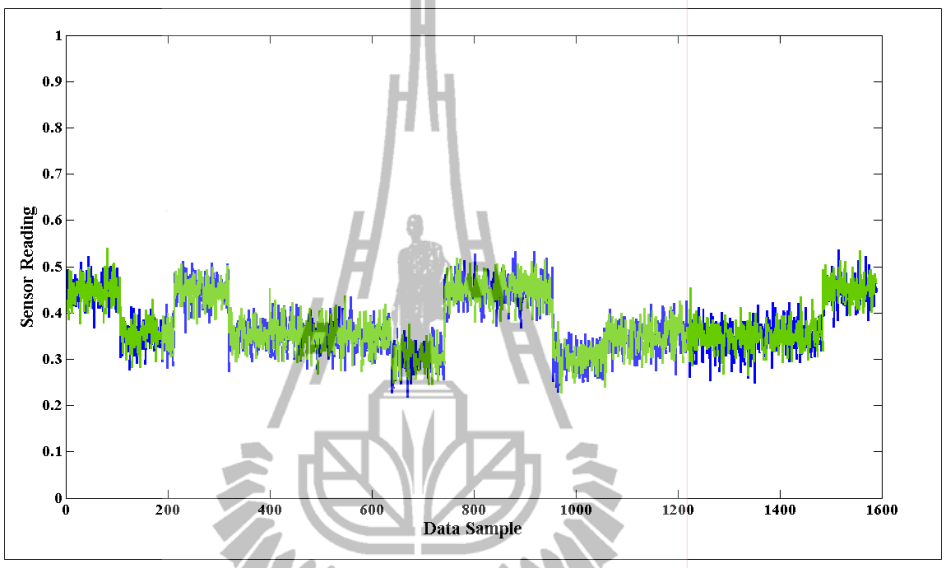


Figure A.1 Synthetic data without fault for training the SOM algorithm.

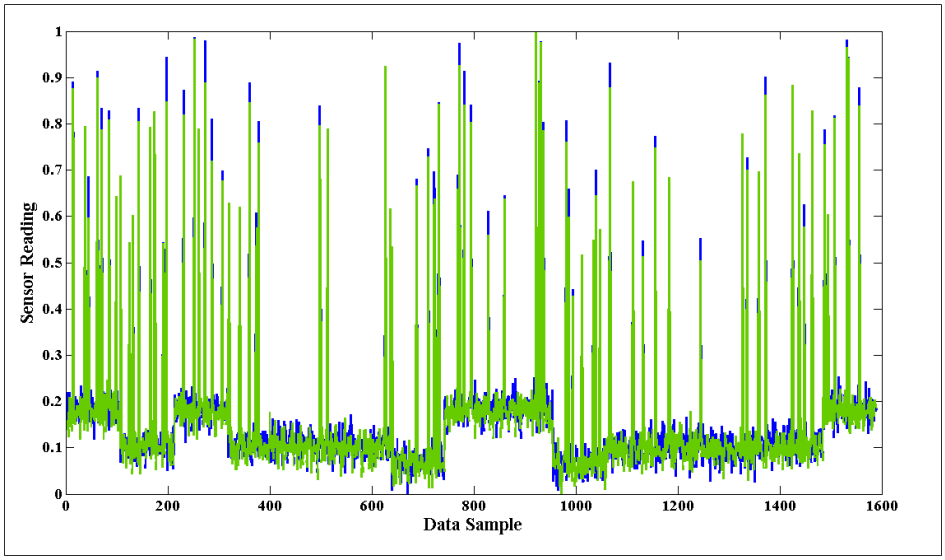


Figure A.2 Synthetic data with 1/80 faults.

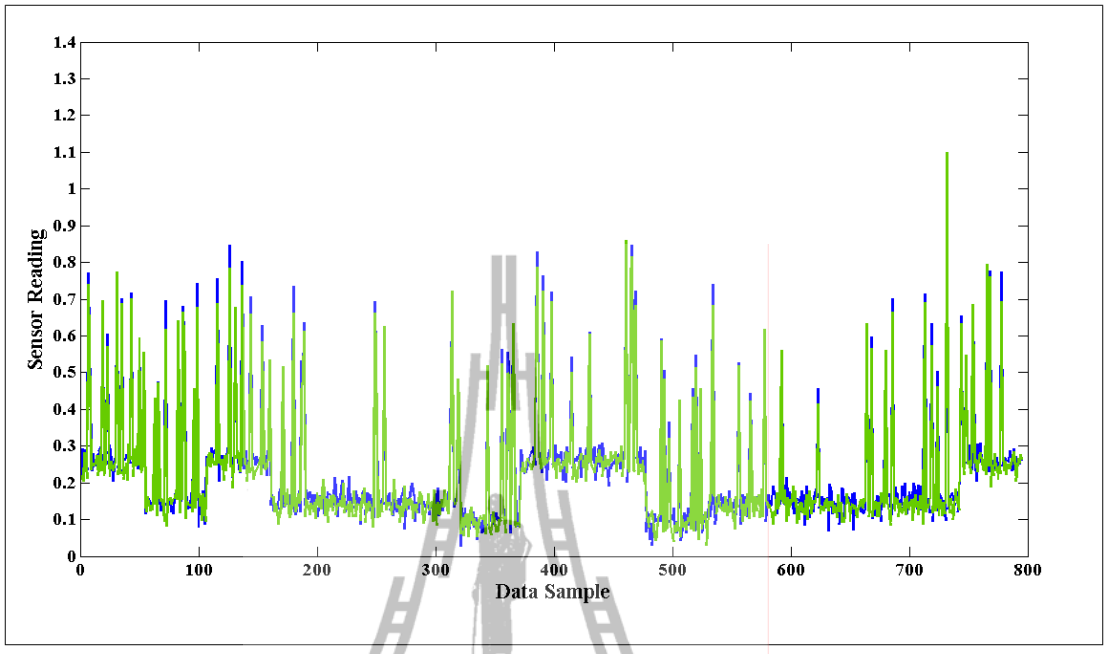


Figure A.3 DWT low-pass coefficient of synthetic data with 1/80 faults.

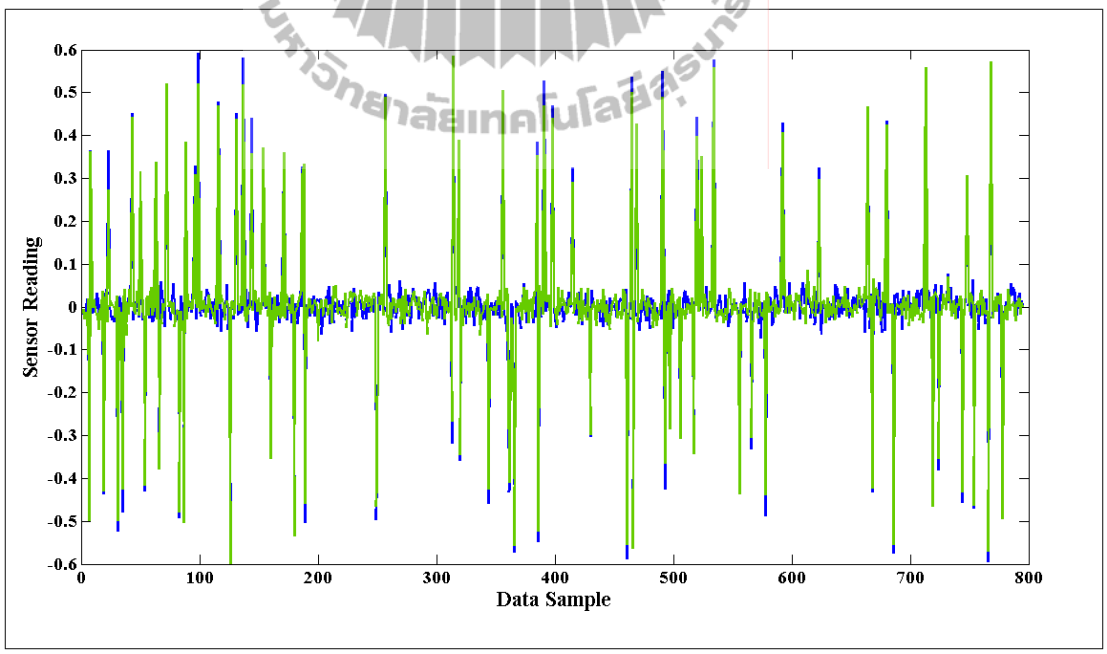


Figure A.4 DWT high-pass coefficient of synthetic data with 1/80 faults.

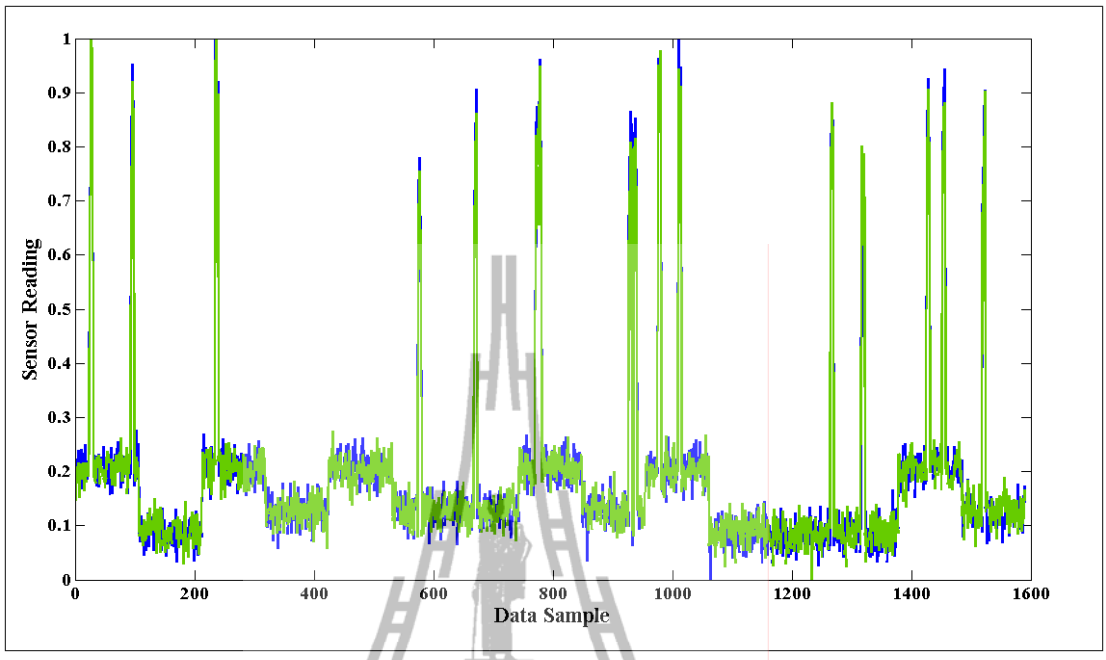


Figure A.5 Synthetic data with 5/16 faults.

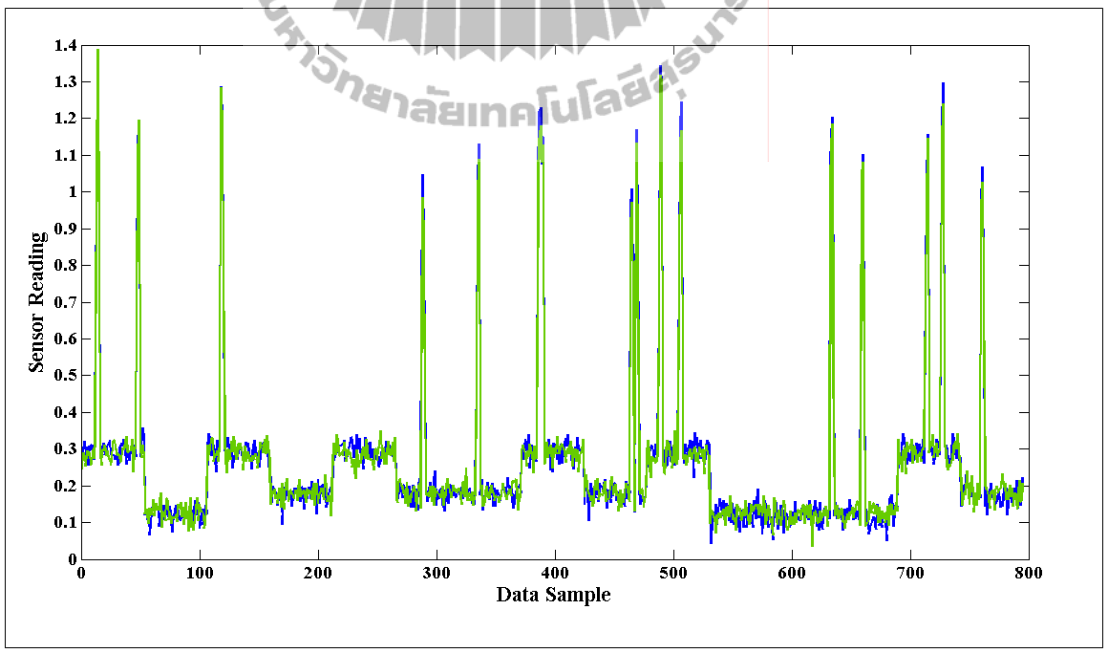


Figure A.6 DWT low-pass coefficient of synthetic data with 5/16 faults.

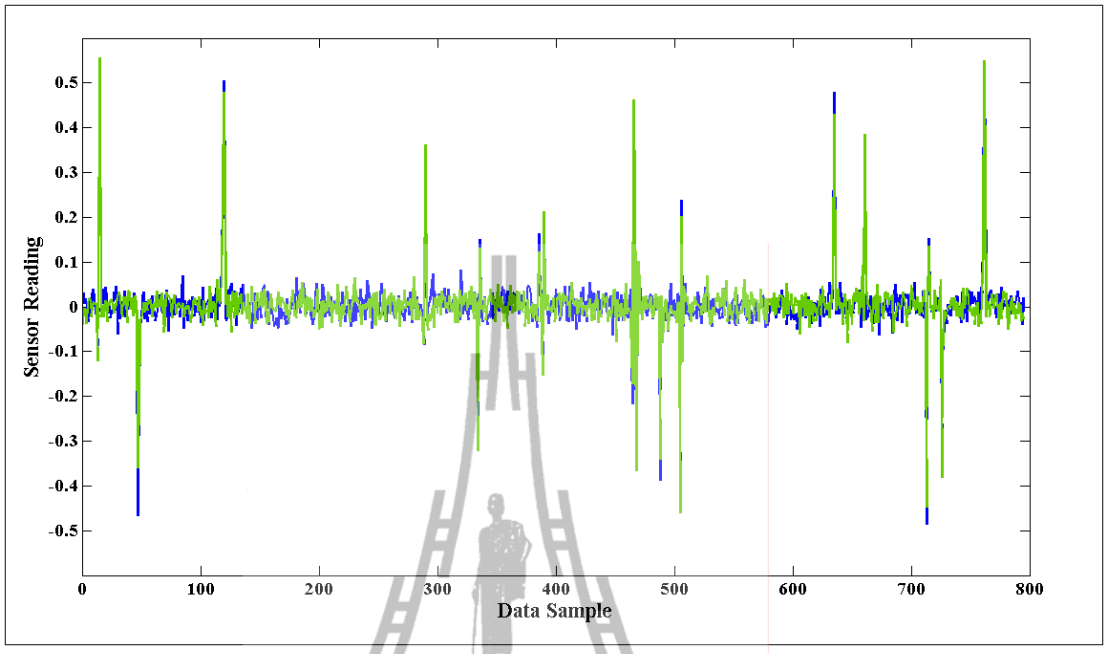


Figure A.7 DWT high-pass coefficient of synthetic data with 5/16 faults.

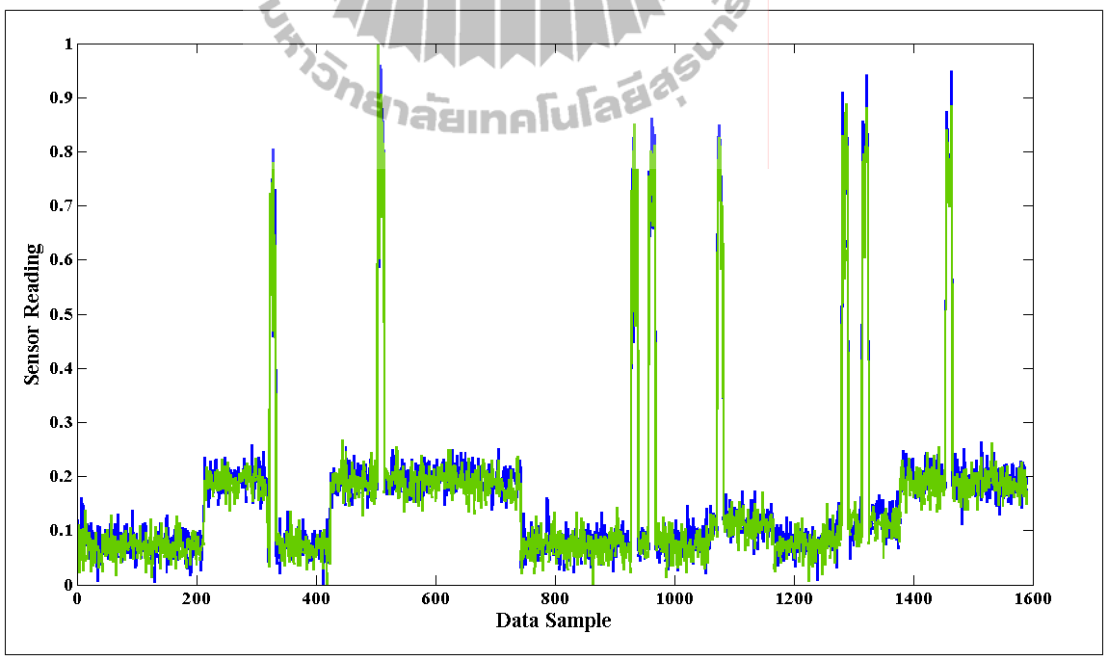


Figure A.8 Synthetic data with 10/8 faults.

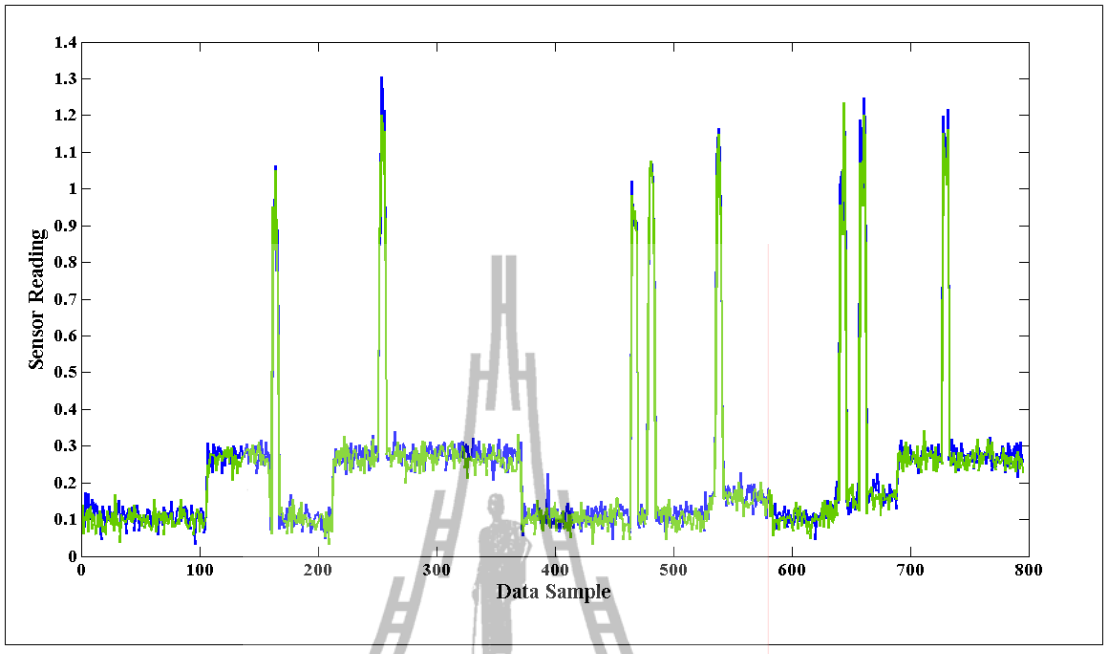


Figure A.9 DWT low-pass coefficient of synthetic data with 10/8 faults.

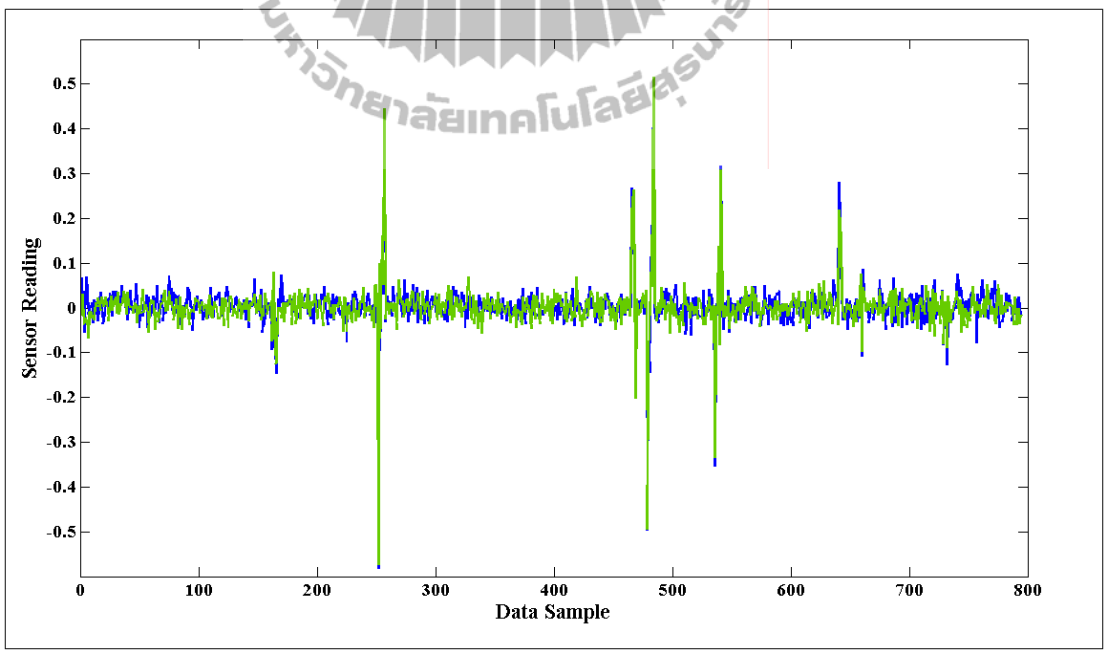


Figure A.10 DWT high-pass coefficient of synthetic data with 10/8 faults.

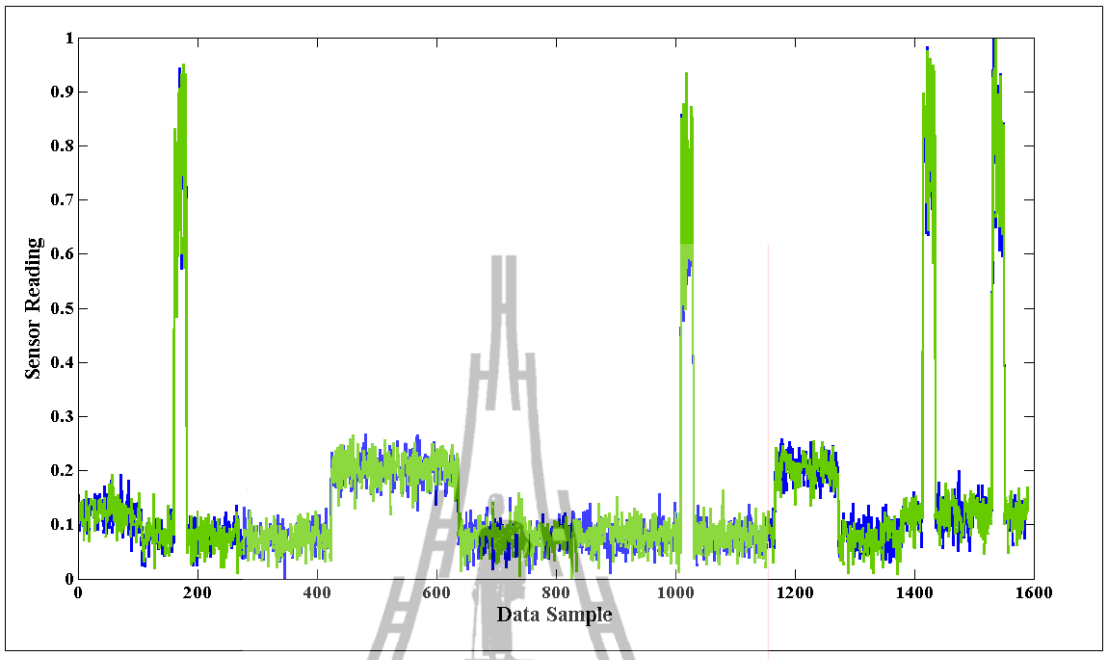


Figure A.11 Synthetic data with 20/4 faults.

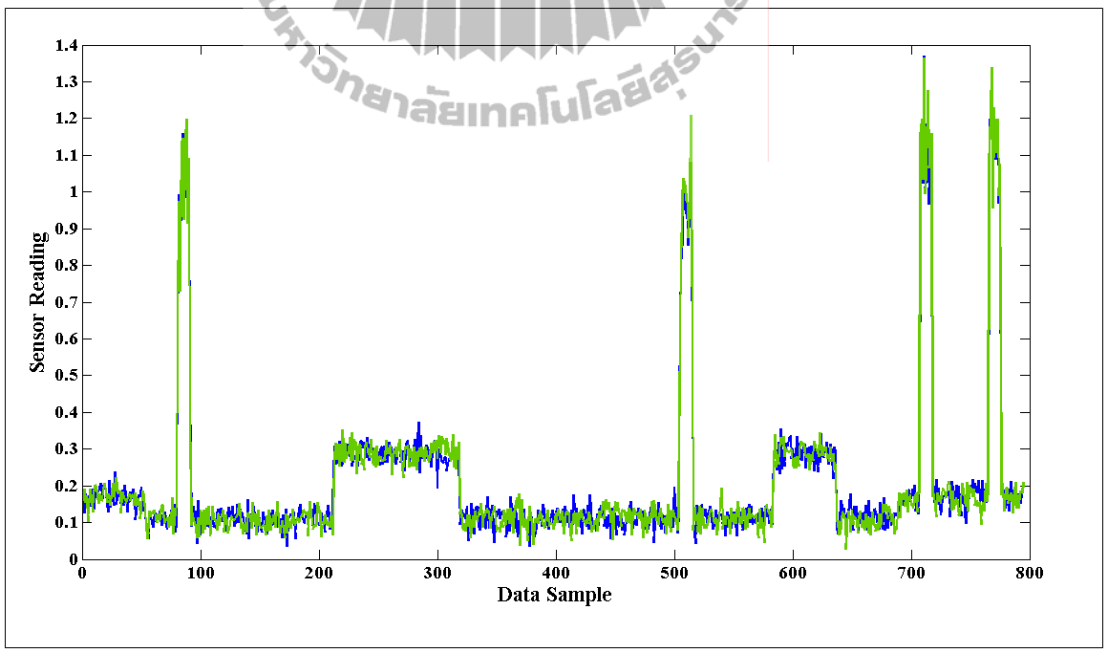


Figure A.12 DWT low-pass coefficient of synthetic data with 20/4 faults.

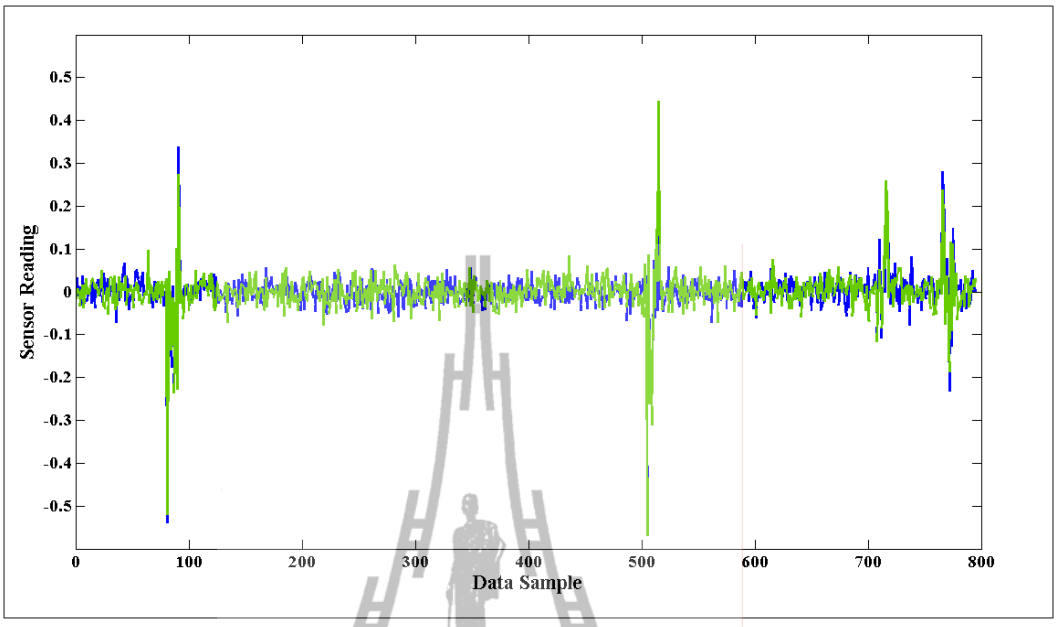


Figure A.13 DWT high-pass coefficient of synthetic data with 20/4 faults.

2. INTEL dataset

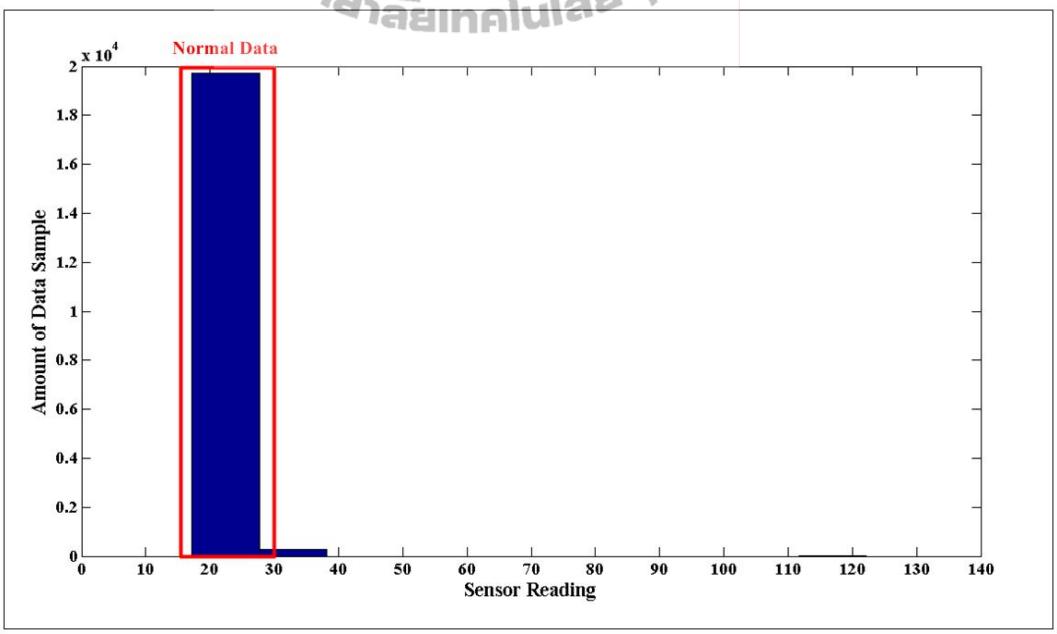


Figure A.14 Histogram of INTEL dataset (temperature reading).

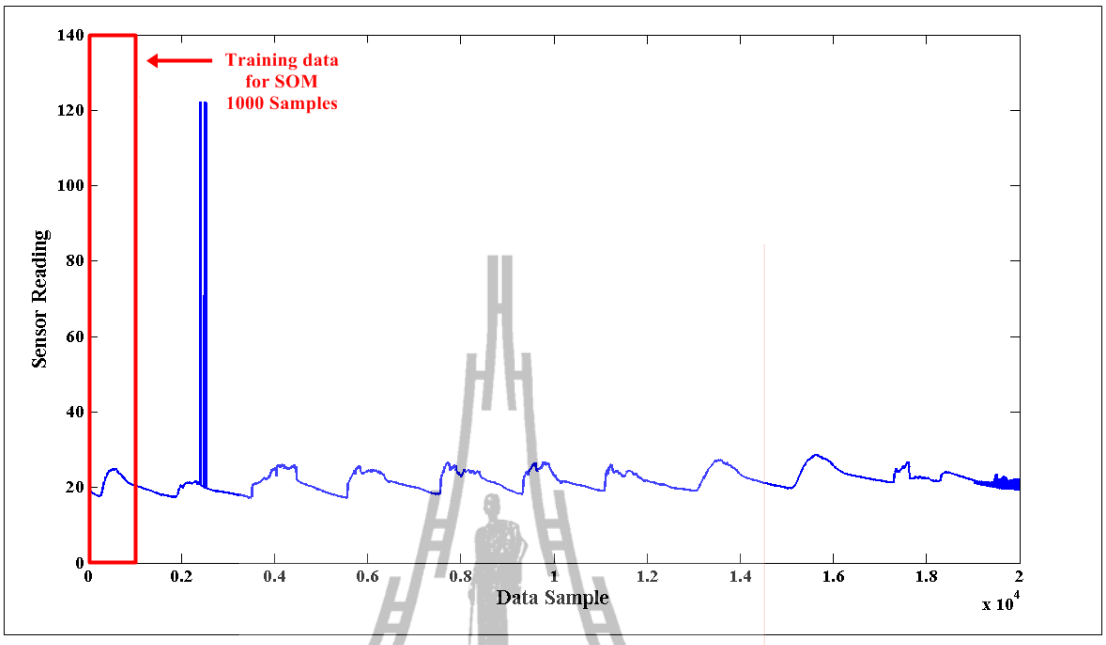


Figure A.15 INTEL dataset (temperature reading).

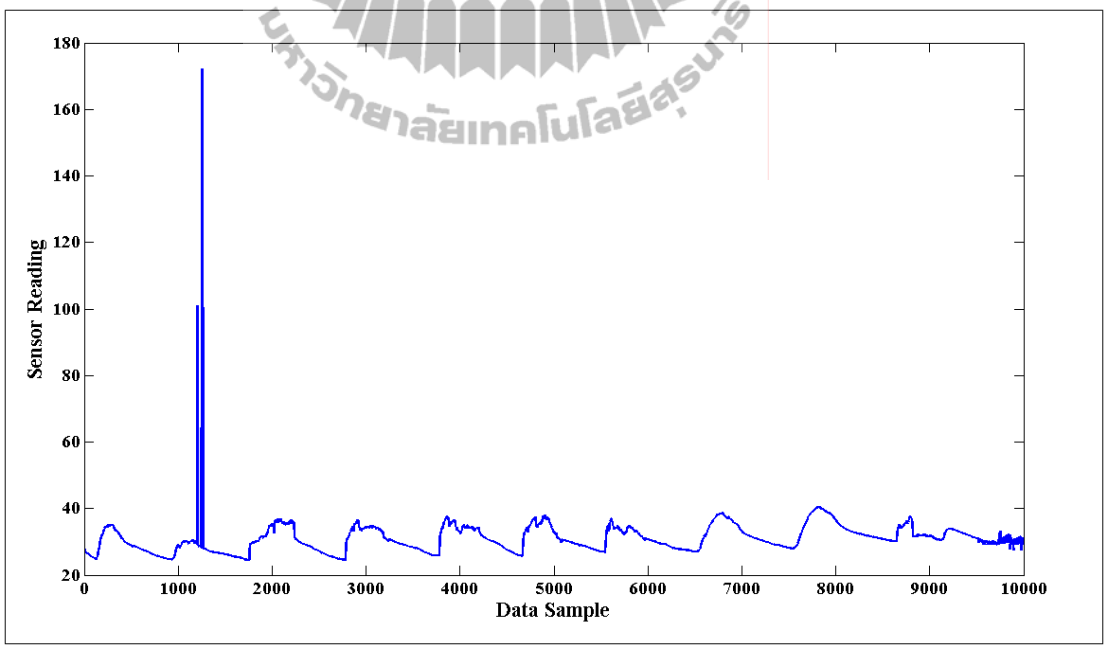


Figure A.16 DWT low-pass coefficient of INTEL dataset (temperature reading).

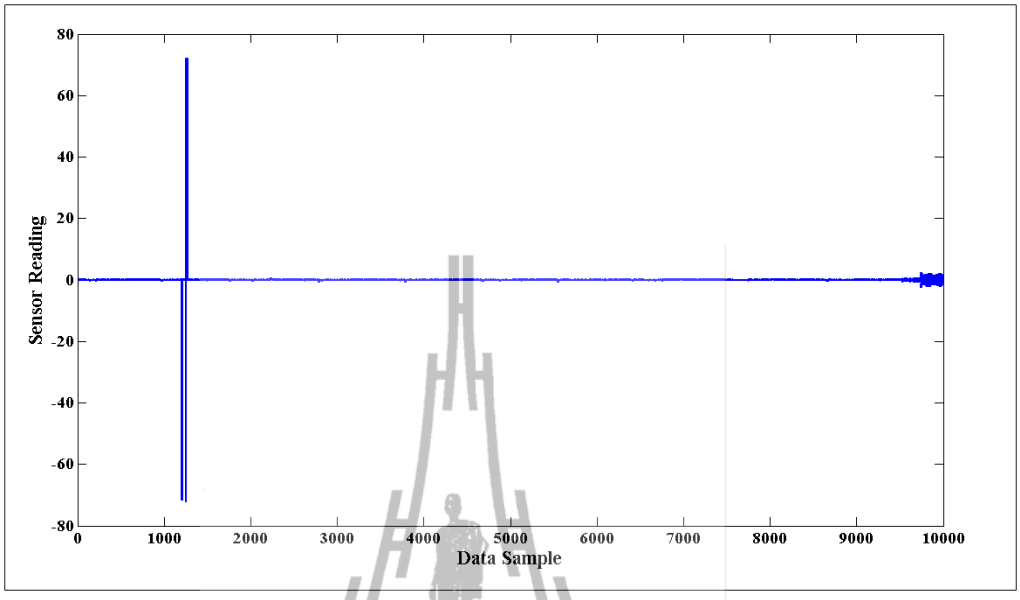


Figure A.17 DWT high-pass coefficient of INTEL dataset (temperature reading).

3. SensorScope Station no.39 dataset (SS39)

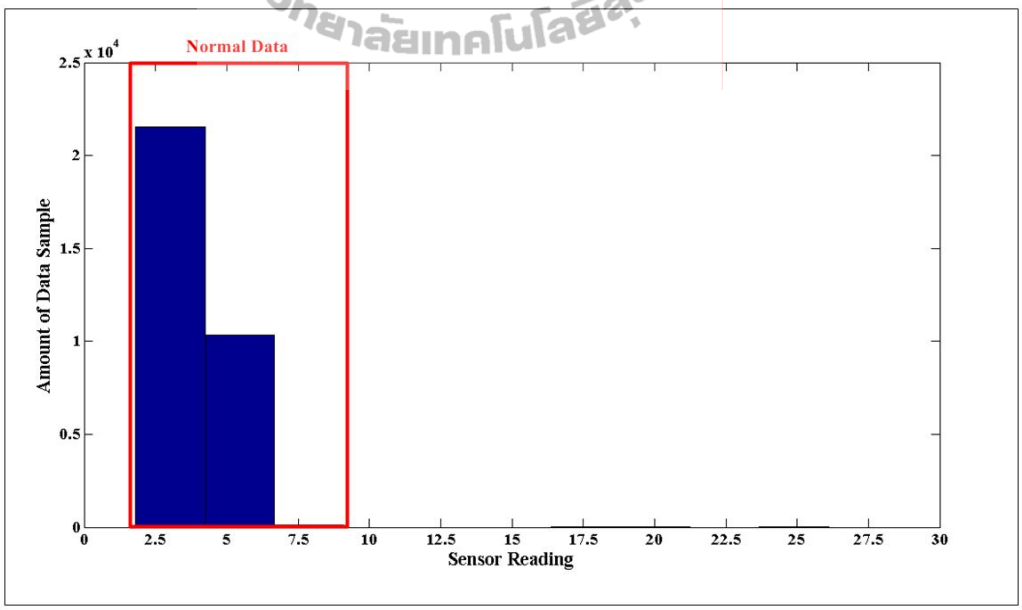


Figure A.18 Histogram of SensorScope Station no.39 (SS39) dataset (global current reading).

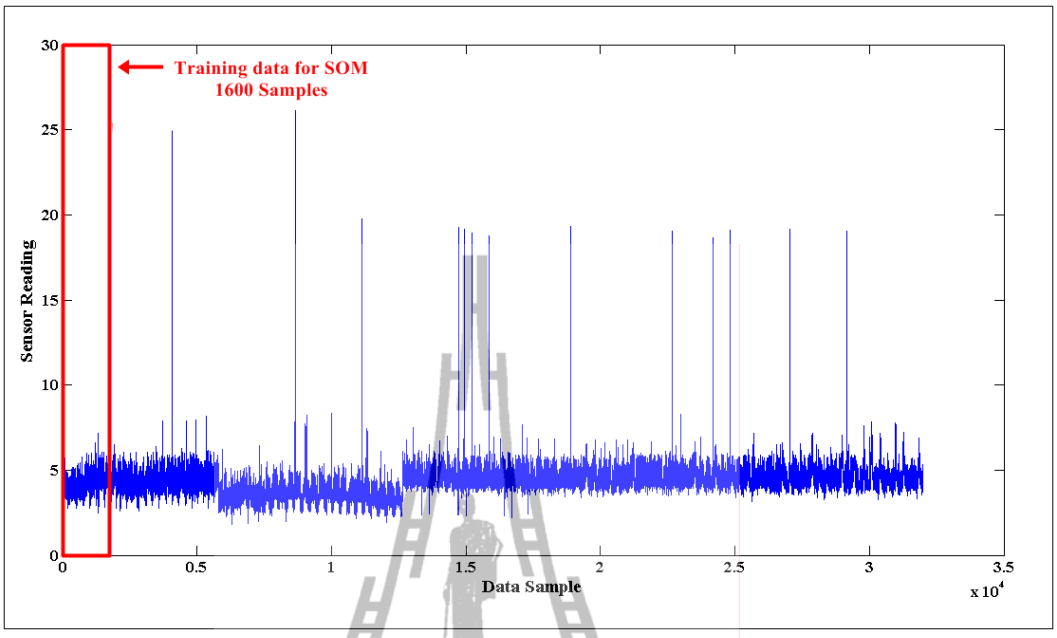


Figure A.19 SensorScope Station no.39 (SS39) dataset (global current reading).

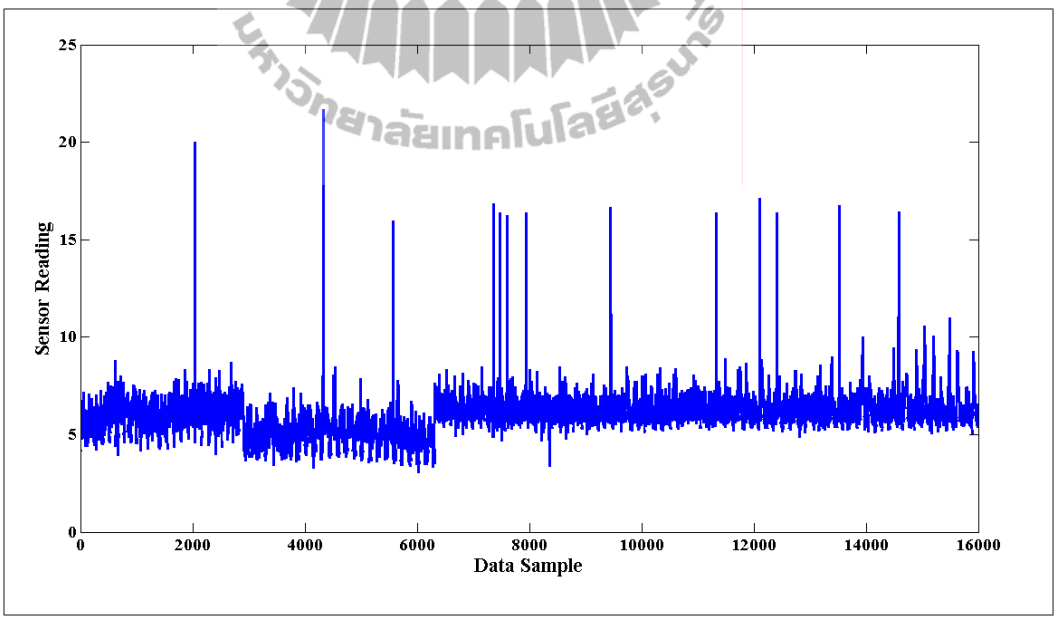


Figure A.20 DWT low-pass coefficient of SensorScope Station no.39 (SS39) dataset (global current reading).

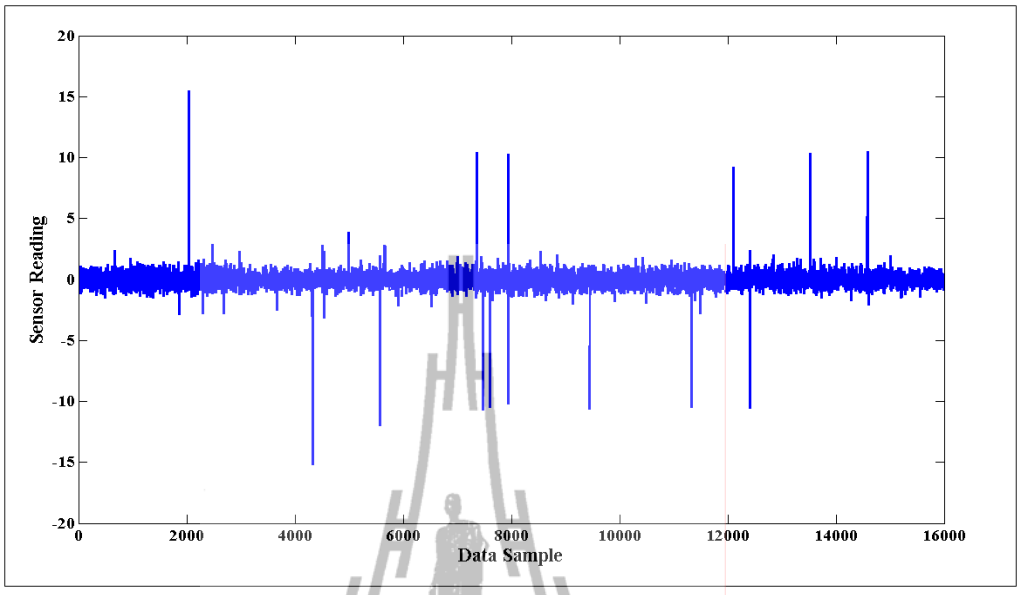


Figure A.21 DWT high-pass coefficient of SensorScope Station no.39 (SS39) dataset (global current reading).

4. SensorScope pdg2008-metro-1 dataset (pdg2008)

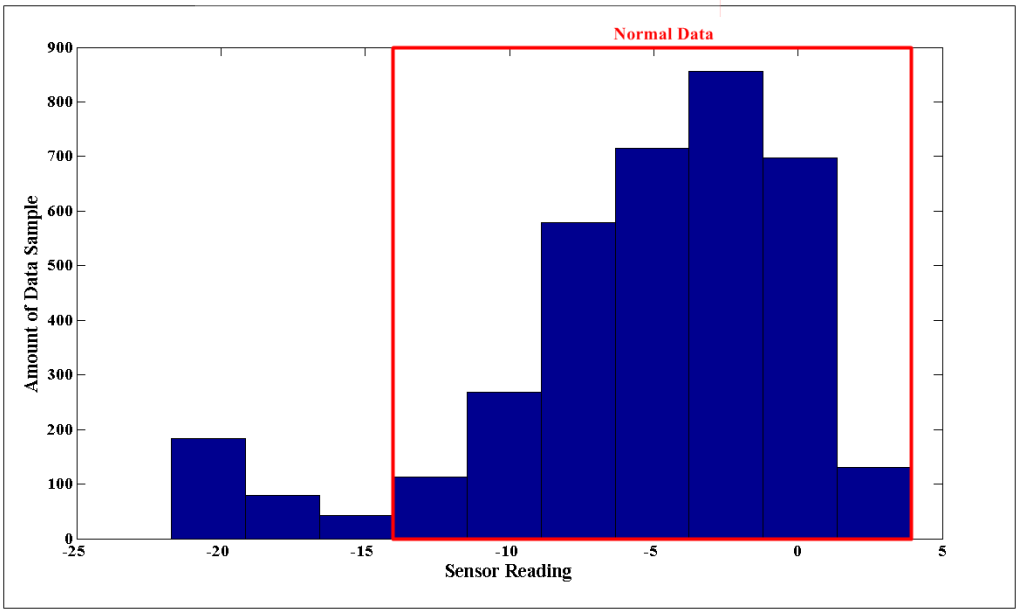


Figure A.22 Histogram of pdg2008 dataset (surface temperature reading).

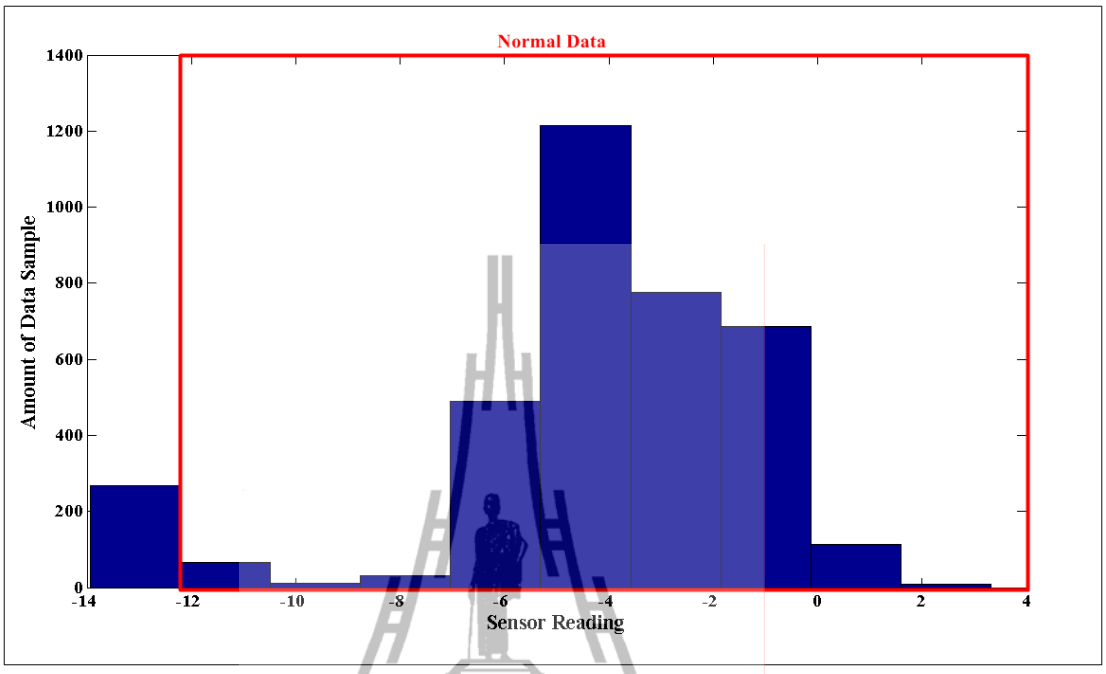


Figure A.23 Histogram of pdg2008 dataset (ambient temperature reading).

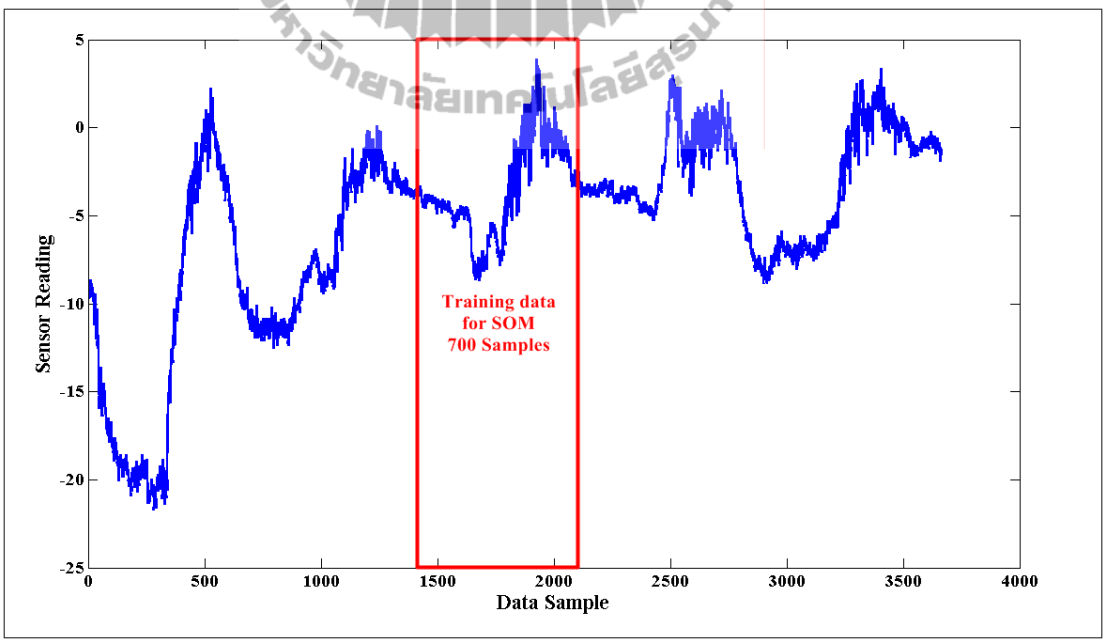


Figure A.24 pdg2008 dataset (surface temperature reading).

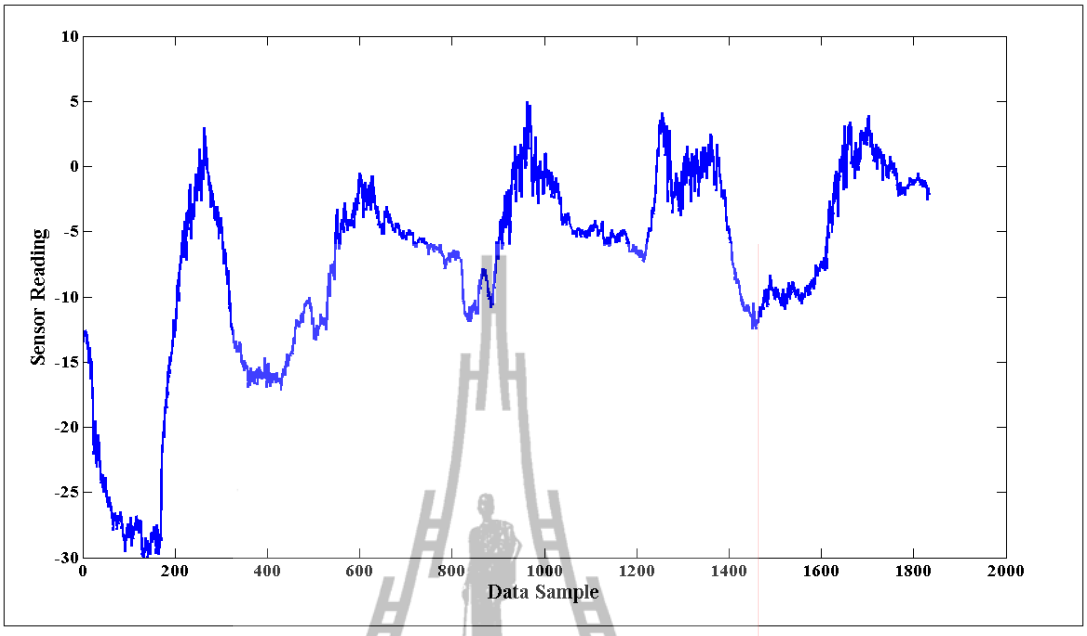


Figure A.25 DWT low-pass coefficient of pdg2008 dataset (surface temperature reading).

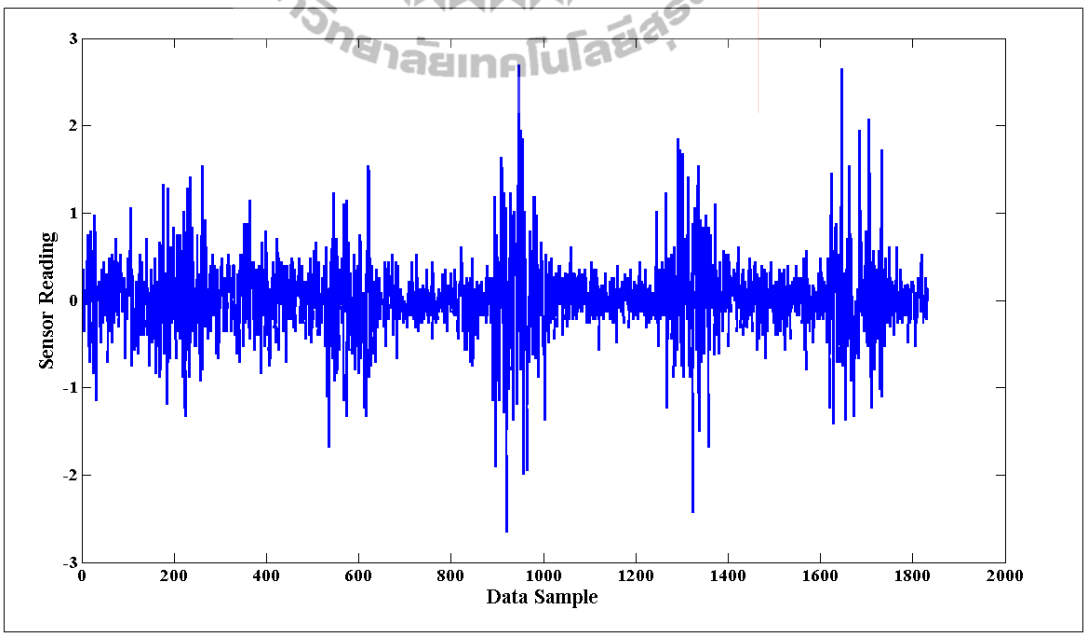


Figure A.26 DWT high-pass coefficient of pdg2008 dataset (surface temperature reading).

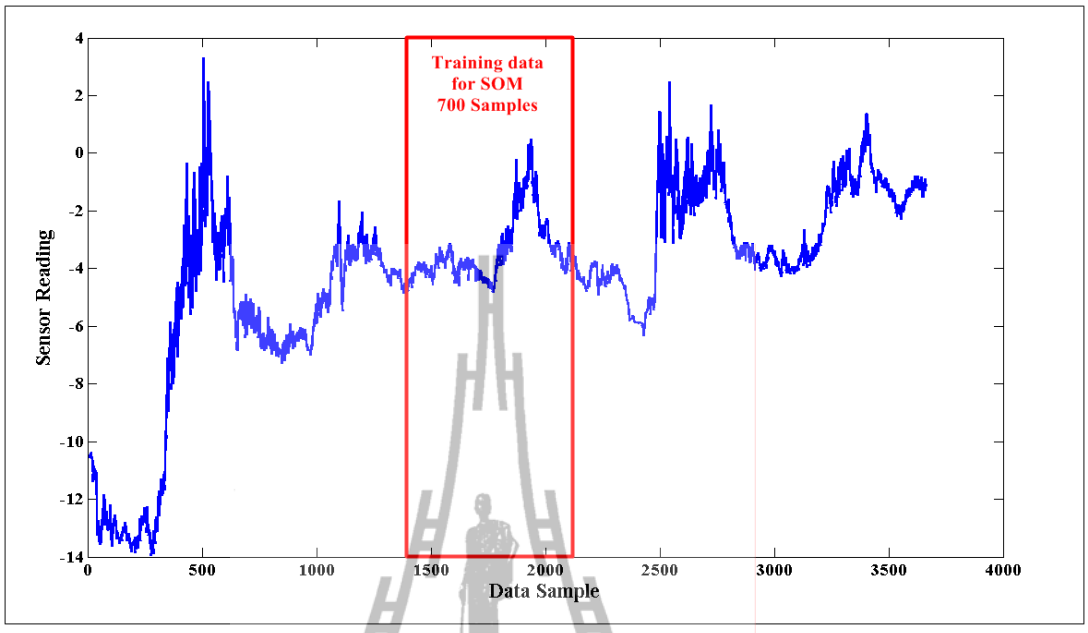


Figure A.27 pdg2008 dataset (ambient temperature reading).

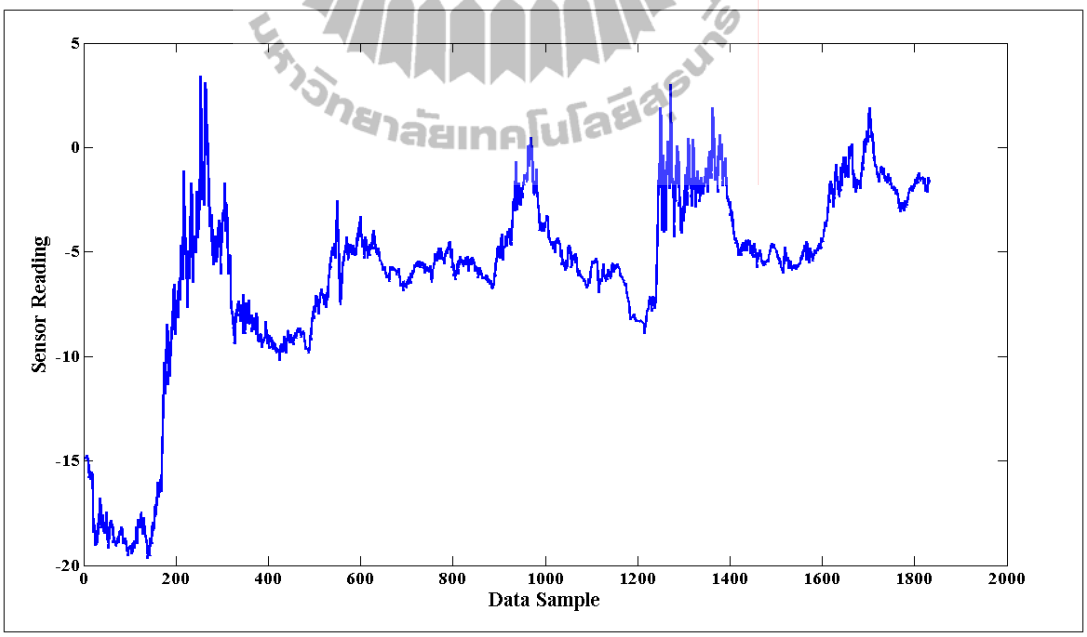


Figure A.28 DWT low-pass coefficient of pdg2008 dataset

(ambient temperature reading).

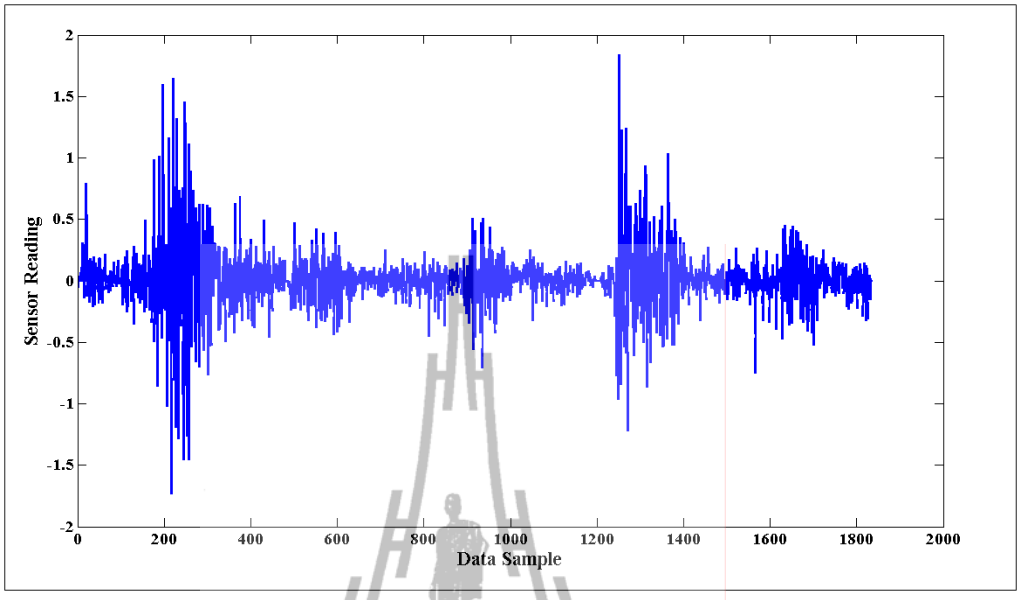


Figure A.29 DWT high-pass coefficient of pdg2008 dataset (ambient temperature reading).

5. NAMOS dataset

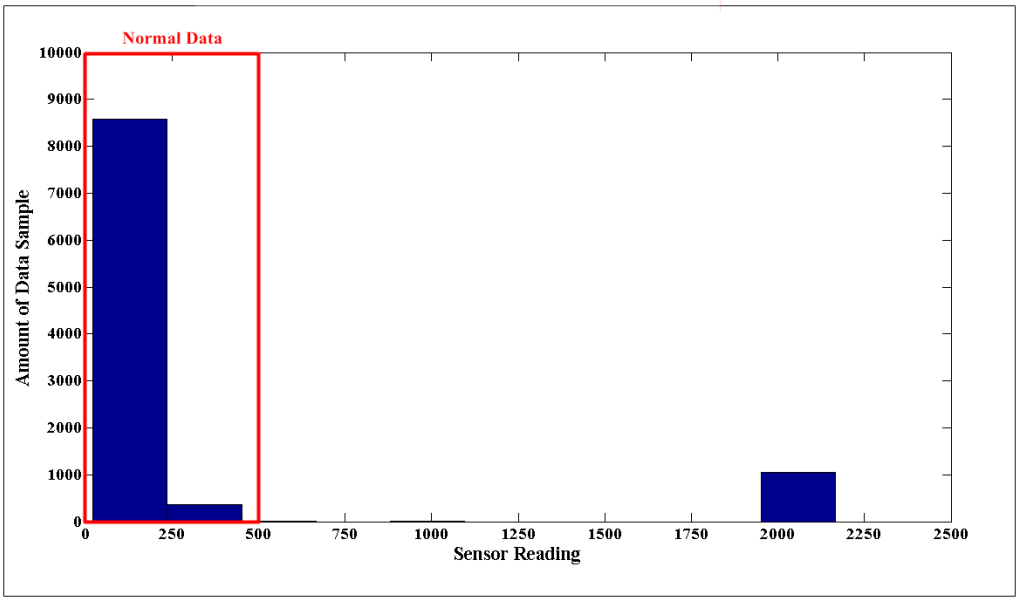


Figure A.30 Histogram of NAMOS dataset (fluorimeters reading).

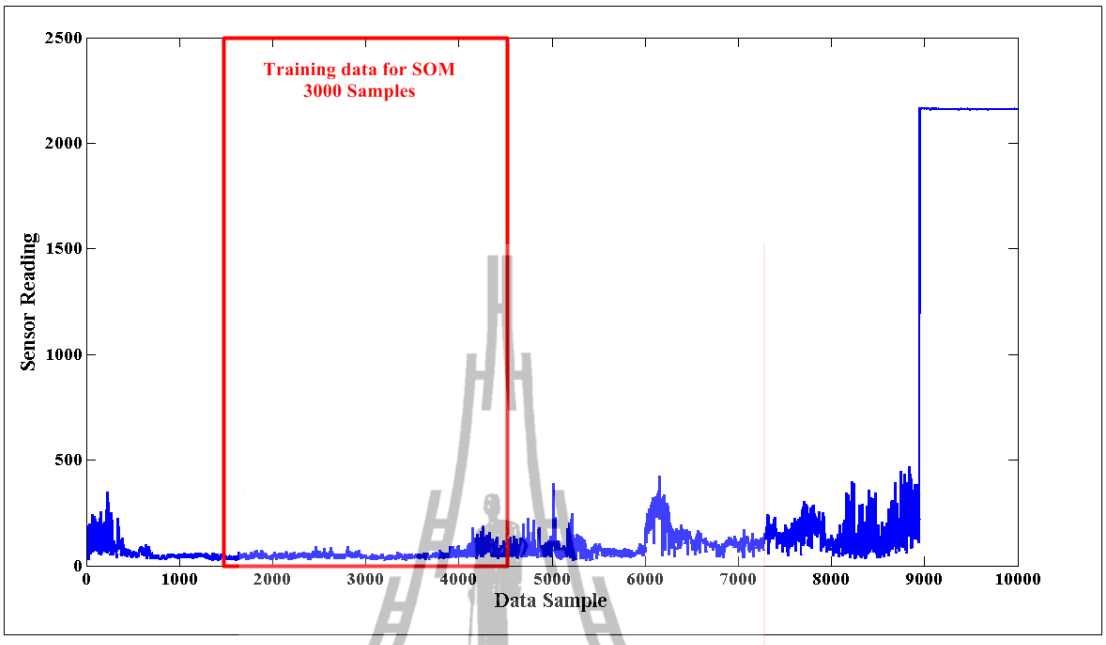


Figure A.31 NAMOS dataset (fluorimeter reading).

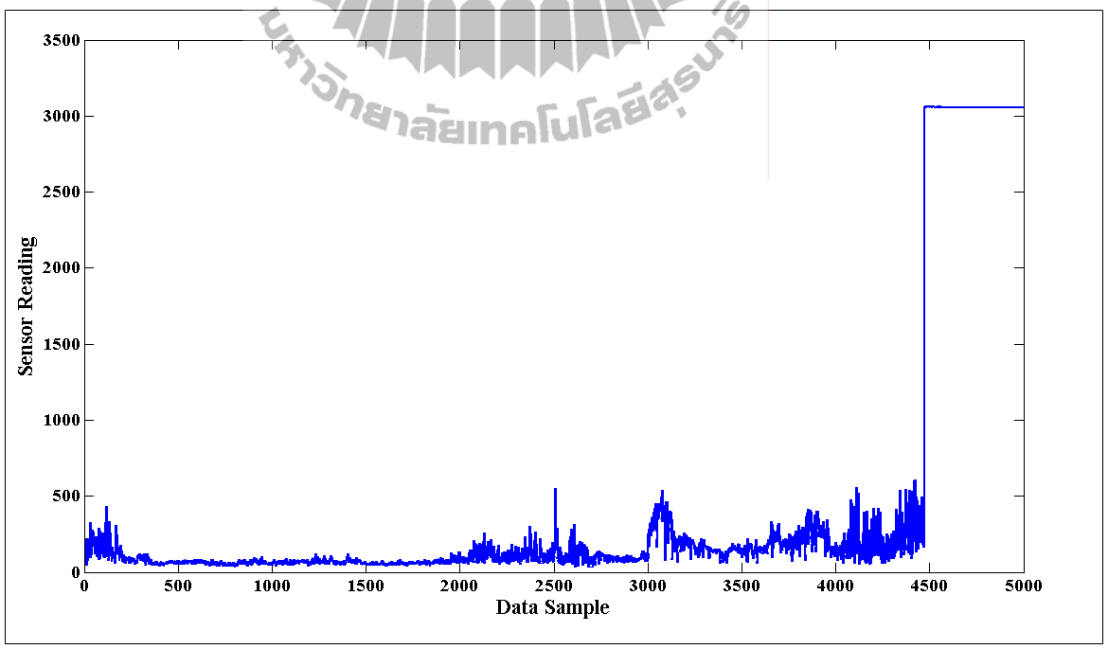


Figure A.32 DWT low-pass coefficient of NAMOS dataset (fluorimeter reading).

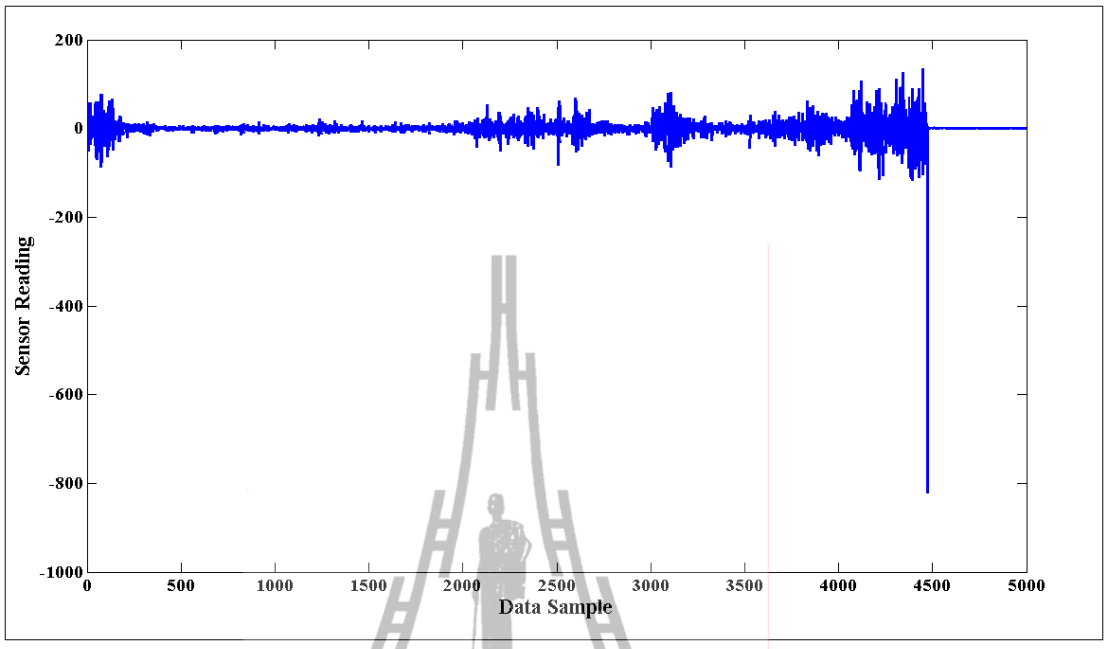


Figure A.33 DWT high-pass coefficient of NAMOS dataset (fluorimeter reading).

APPENDIX B

DATASETS FOR EXPERIMENT CHAPTER 4

Datasets for Chapter 4

1. Synthetic Data

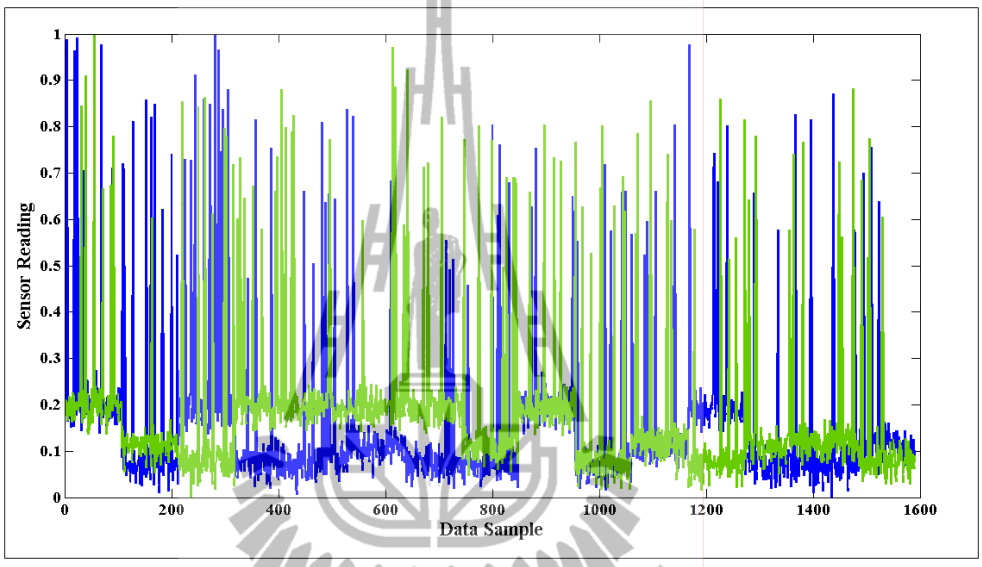


Figure B.1 2KPI Synthetic data with 1/80 faults.

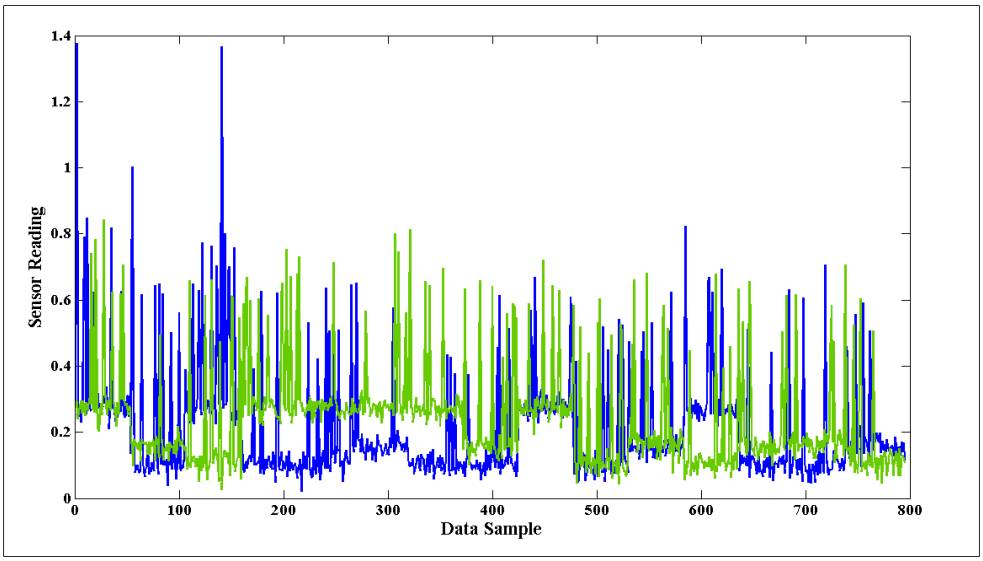


Figure B.2 DWT low-pass coefficient of 2KPI synthetic data with 1/80 faults.

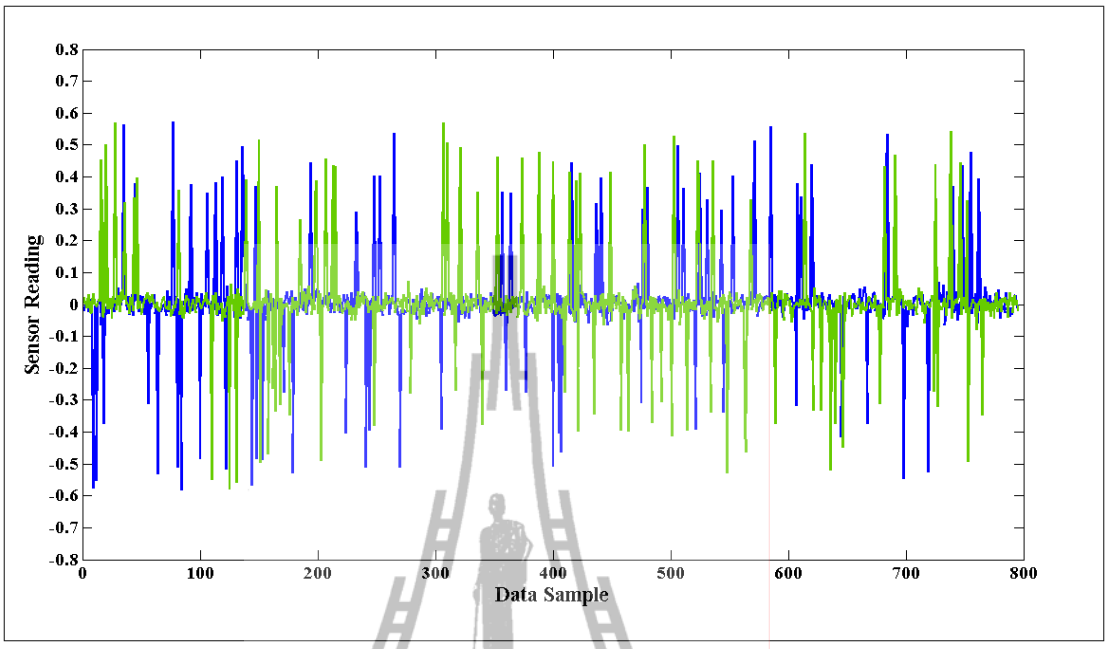


Figure B.3 DWT high-pass coefficient of 2KPI synthetic data with 1/80 faults.

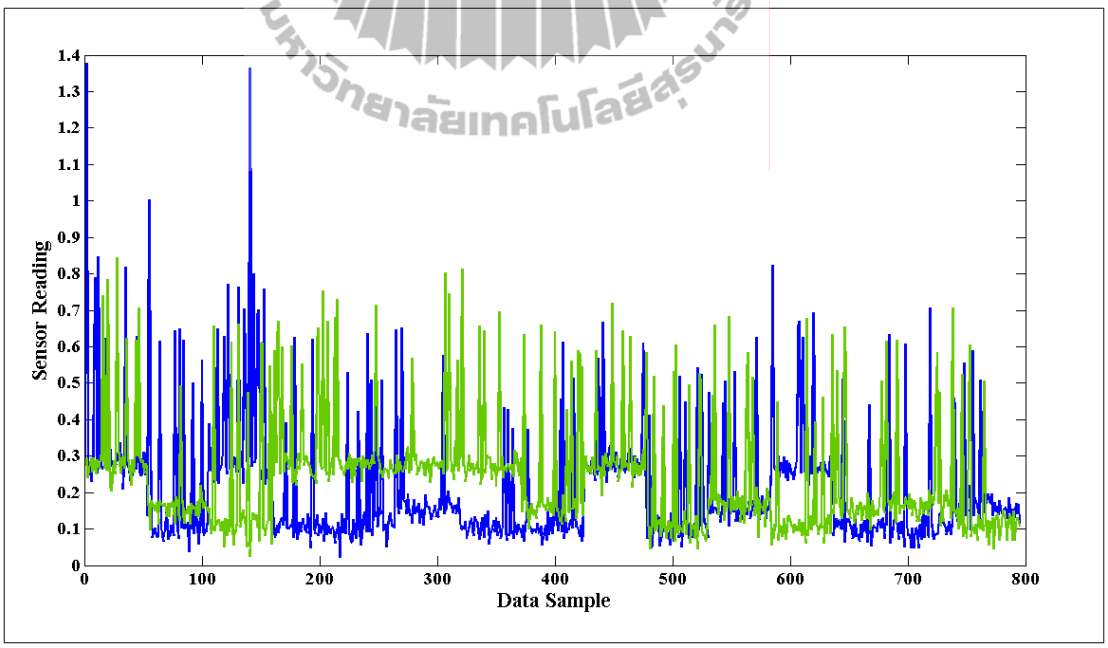


Figure B.4 LWT low-pass coefficient of 2KPI synthetic data with 1/80 faults.

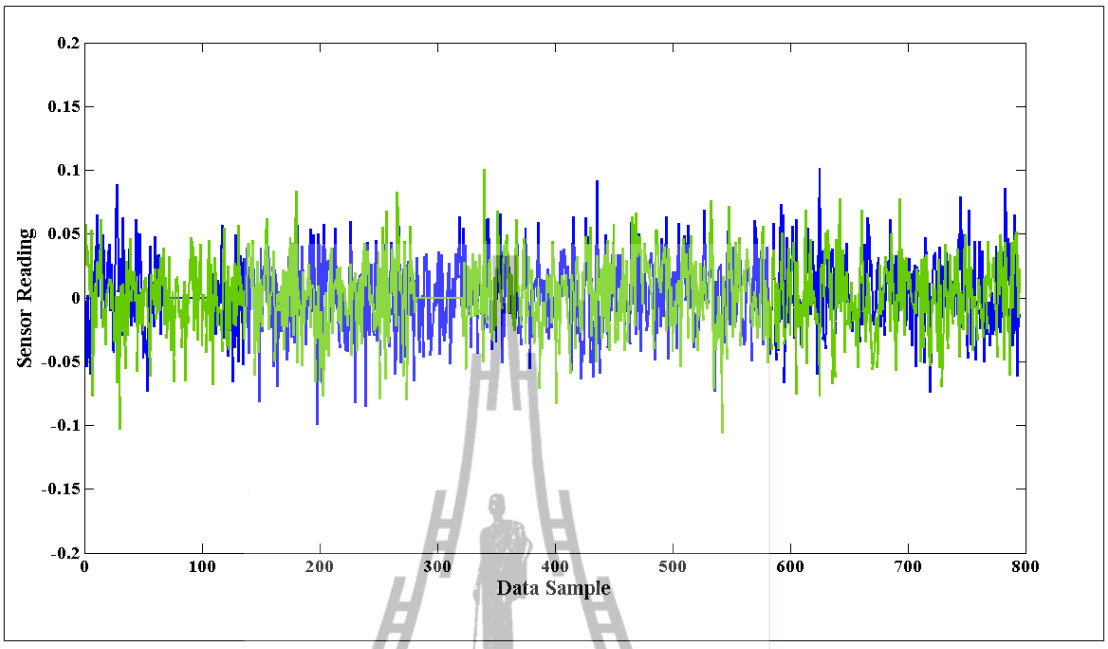


Figure B.5 LWT high-pass coefficient of 2KPI synthetic data with 1/80 faults.

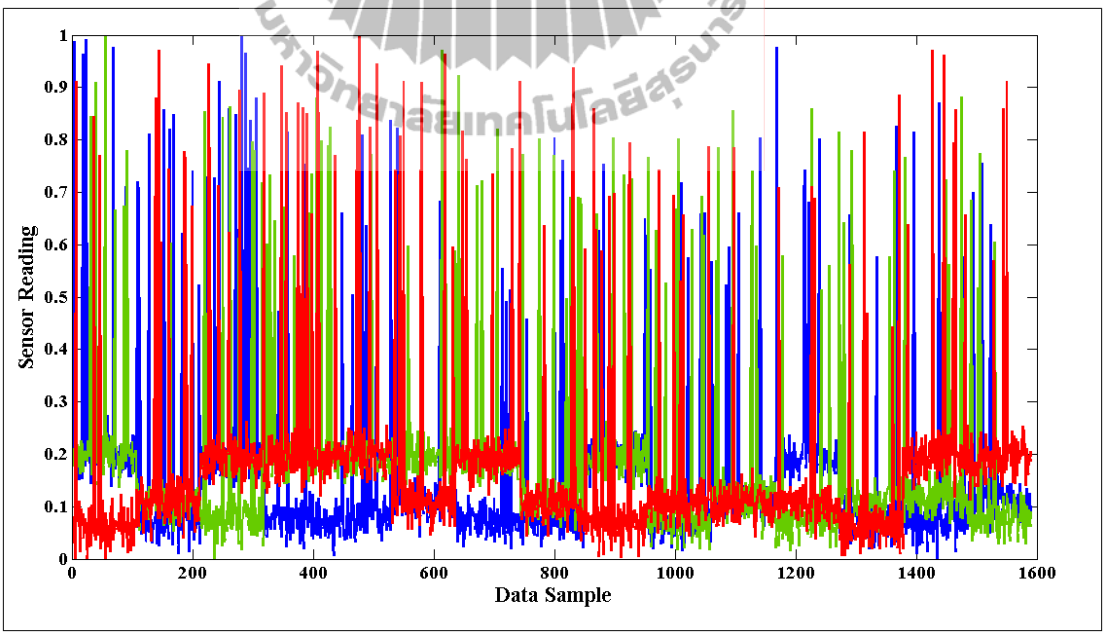


Figure B.6 3KPI Synthetic data with 1/80 faults.

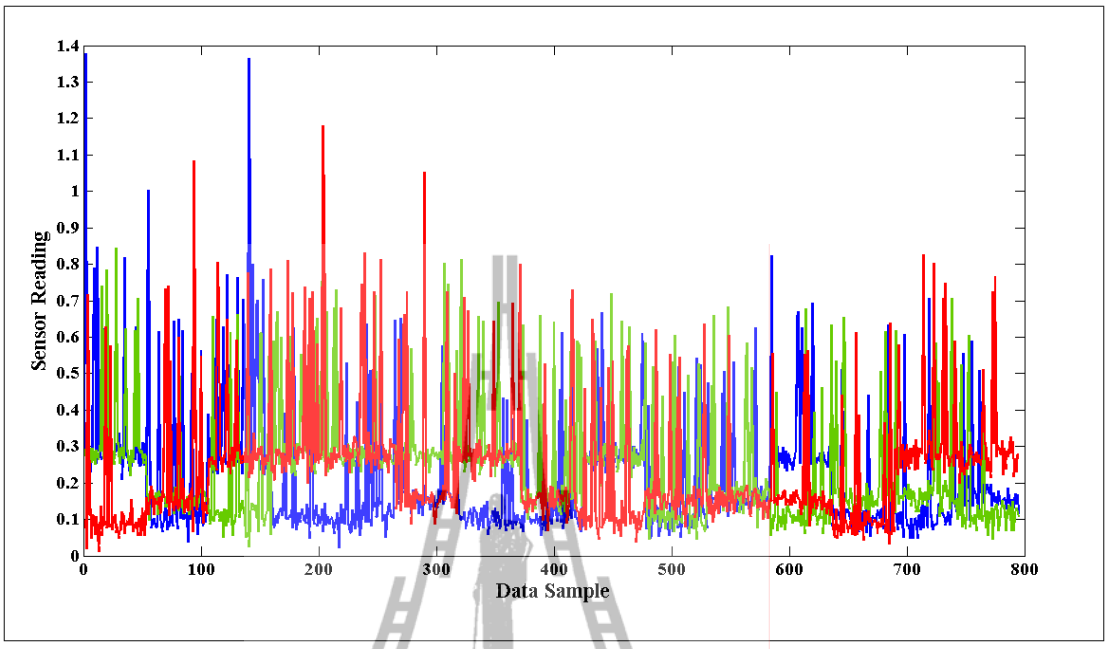


Figure B.7 DWT low-pass coefficient of 3KPI synthetic data with 1/80 faults.

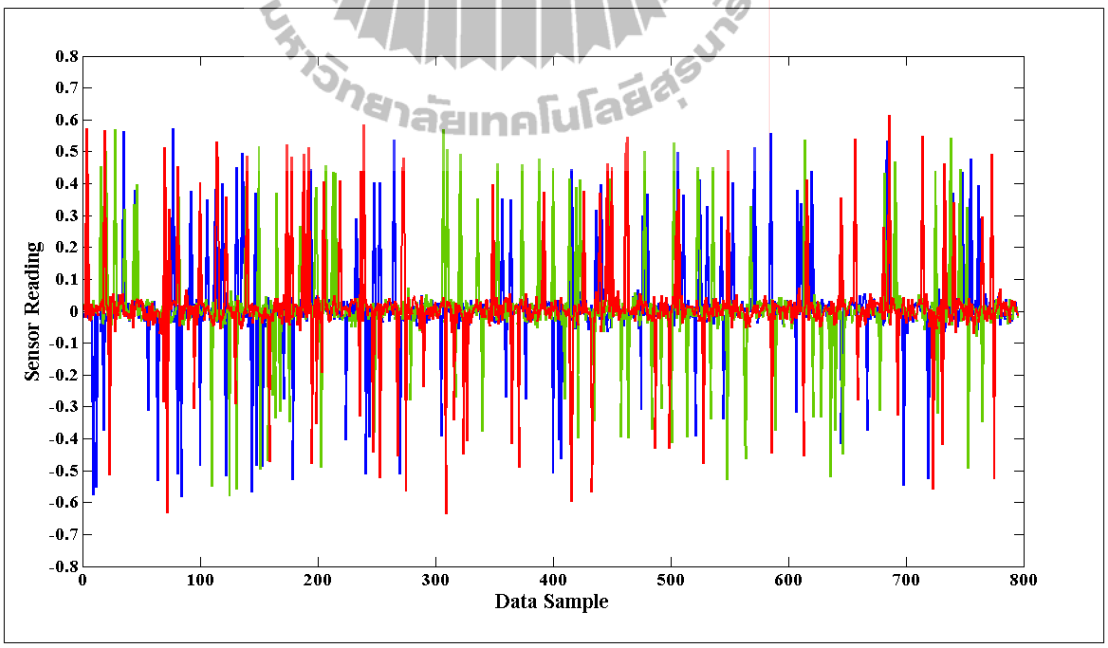


Figure B.8 DWT high-pass coefficient of 3KPI synthetic data with 1/80 faults.

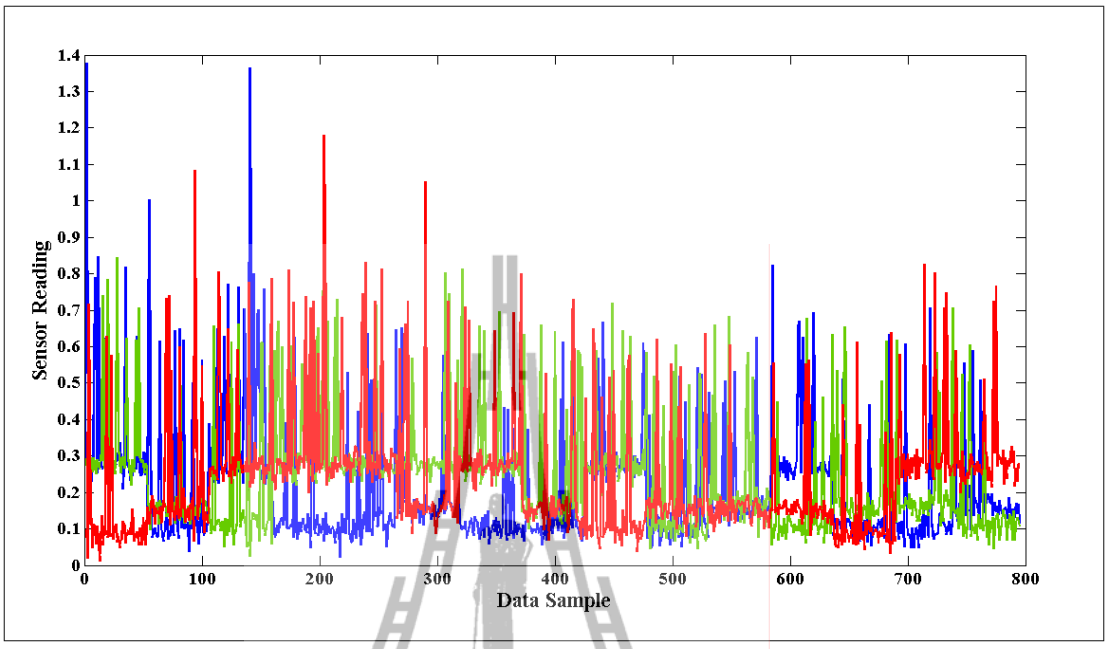


Figure B.9 LWT low-pass coefficient of 3KPI synthetic data with 1/80 faults.

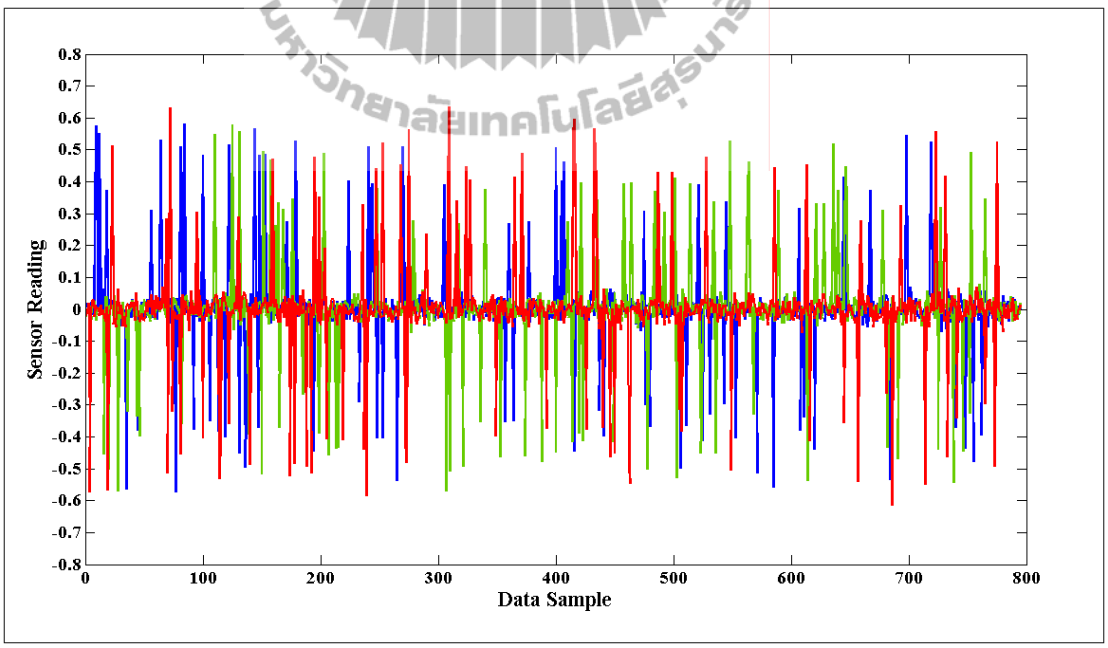


Figure B.10 LWT high-pass coefficient of 3KPI synthetic data with 1/80 faults.

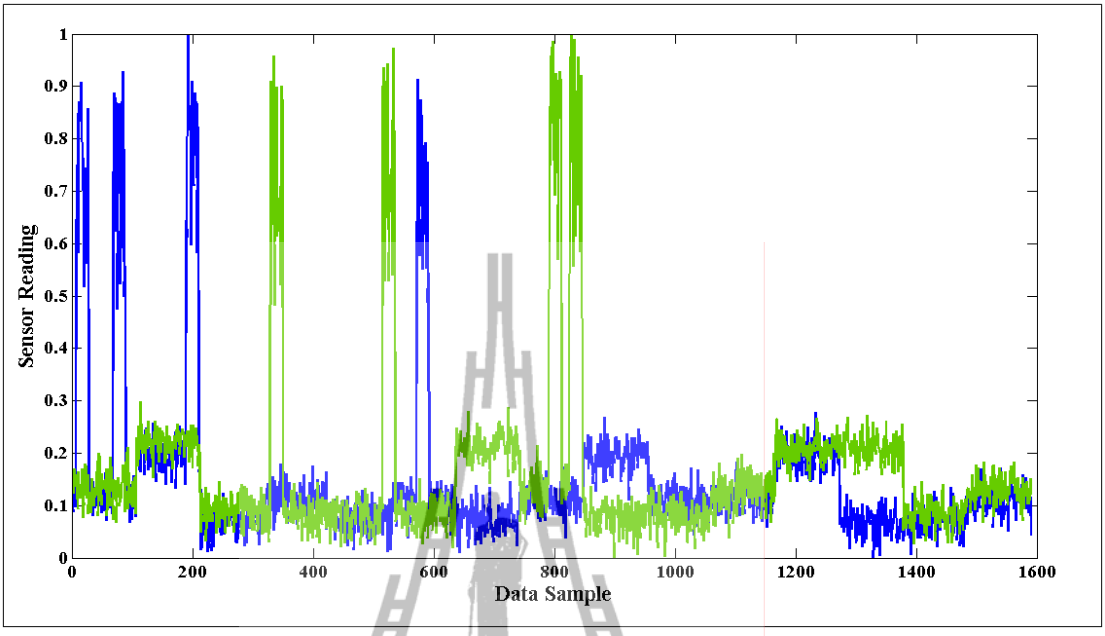


Figure B.11 2KPI Synthetic data with 20/4 faults.

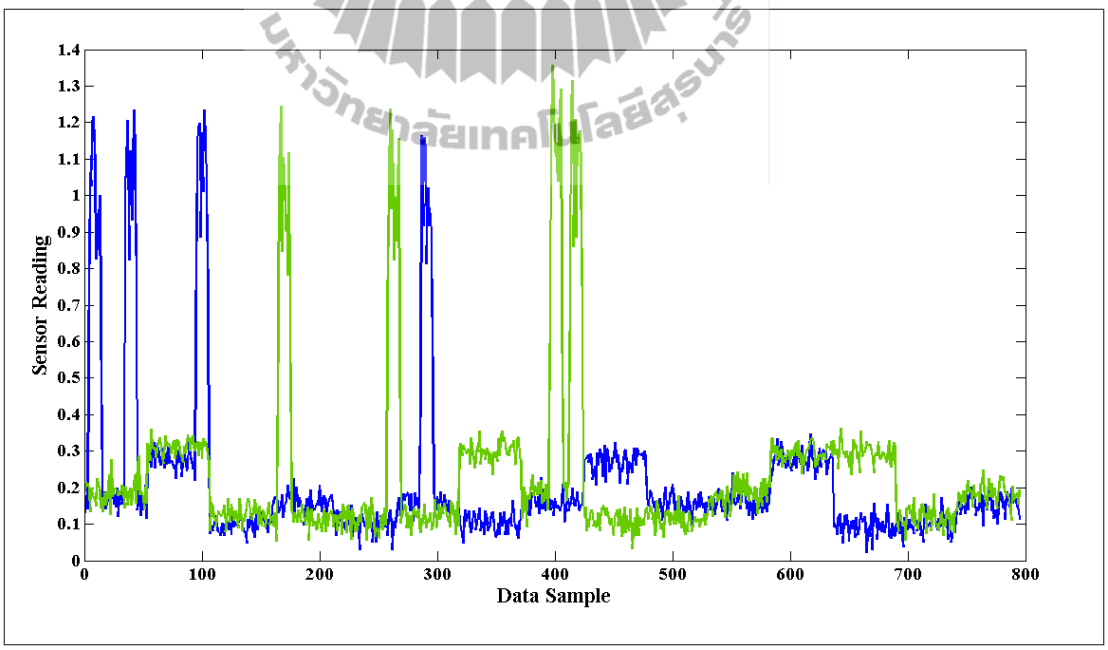


Figure B.12 DWT low-pass coefficient of 2KPI synthetic data with 20/4 faults.

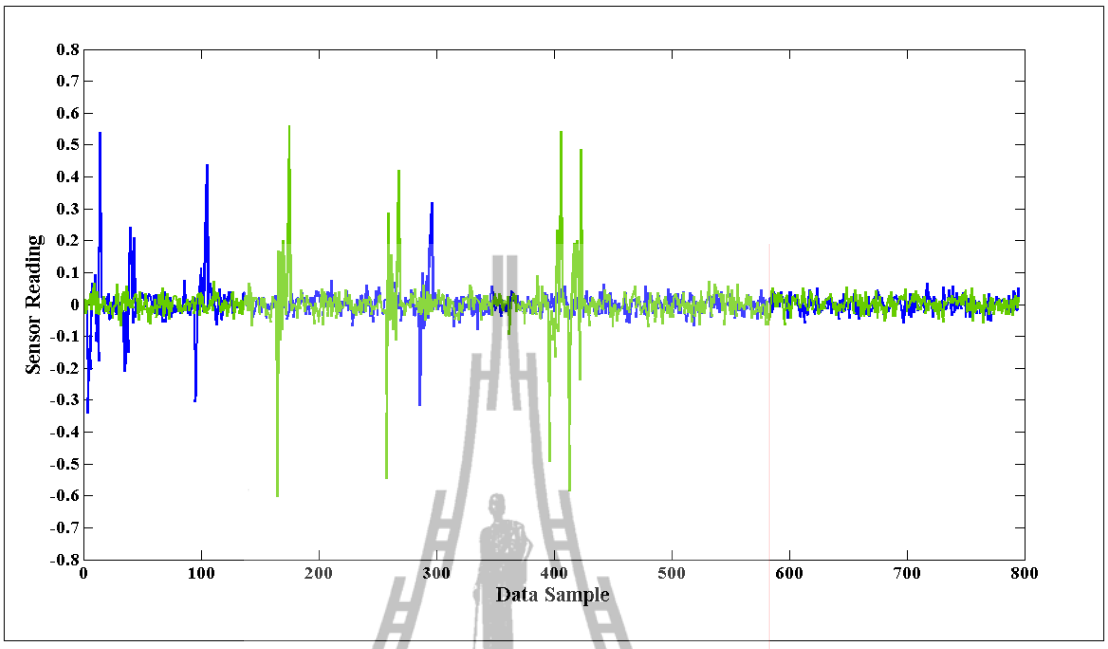


Figure B.13 DWT high-pass coefficient of 2KPI synthetic data with 20/4 faults.

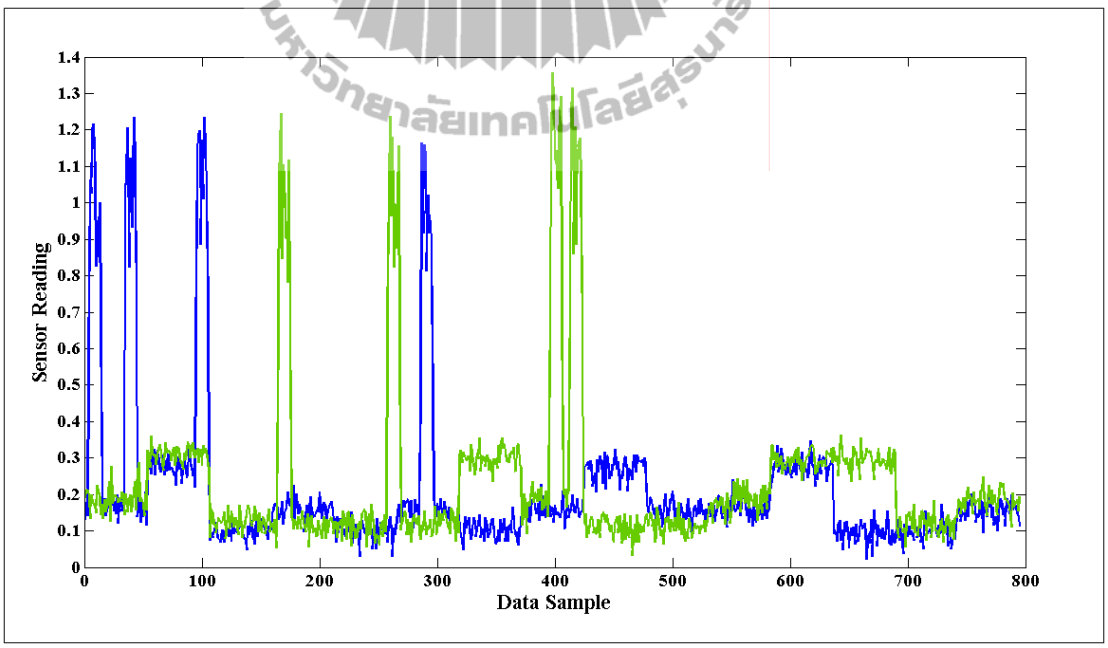


Figure B.14 LWT low-pass coefficient of 2KPI synthetic data with 20/4 faults.

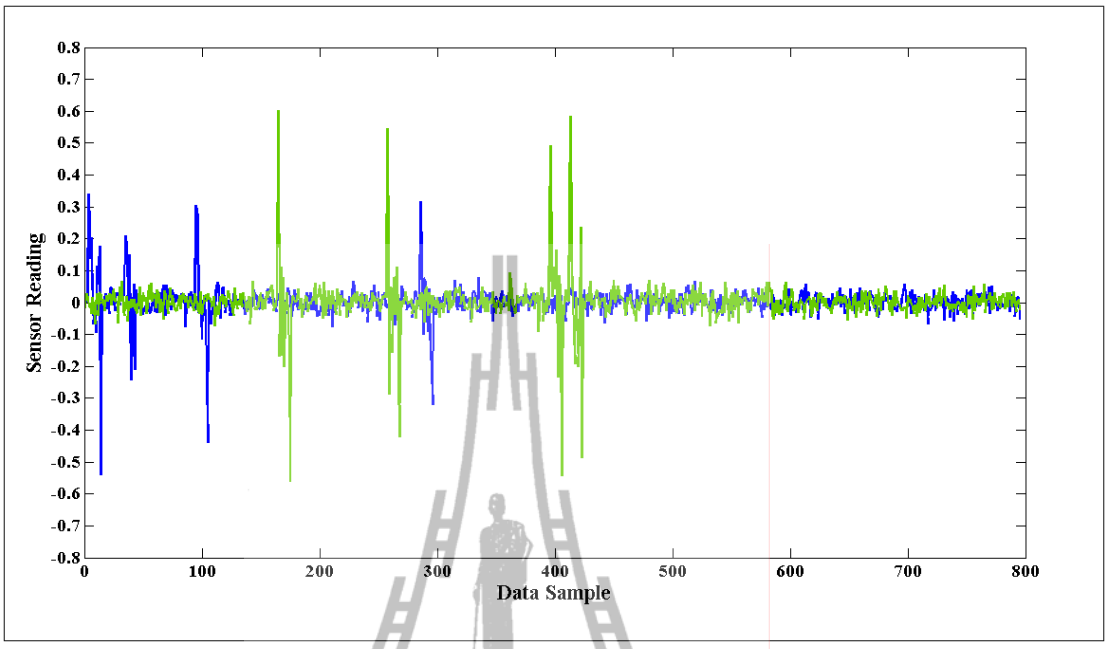


Figure B.15 LWT high-pass coefficient of 2KPI synthetic data with 20/4 faults.

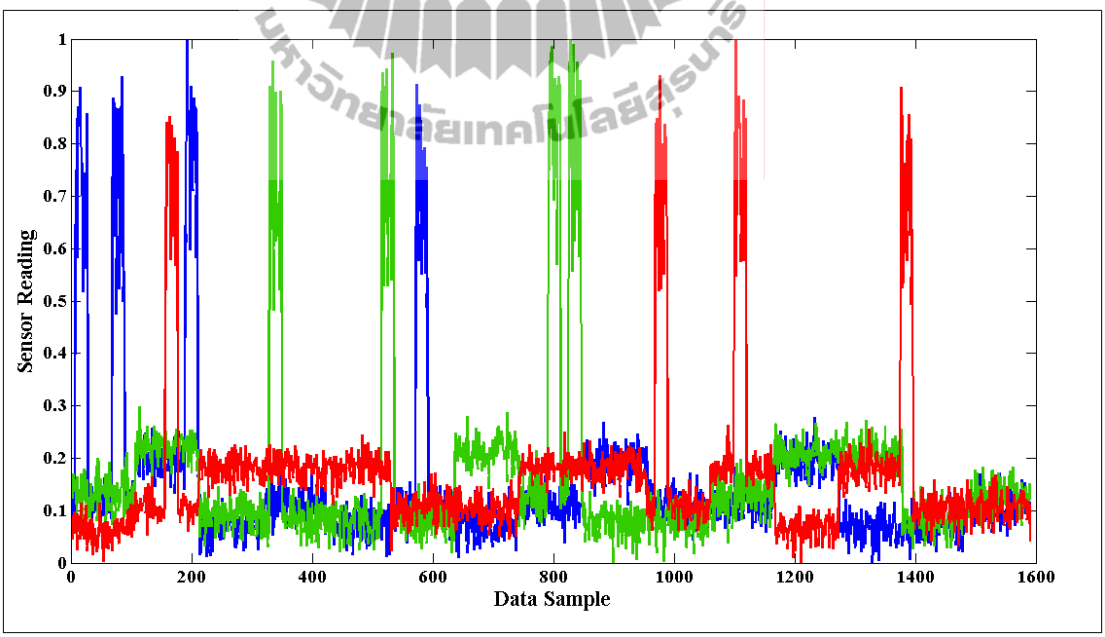


Figure B.16 3KPI Synthetic data with 20/4 faults.

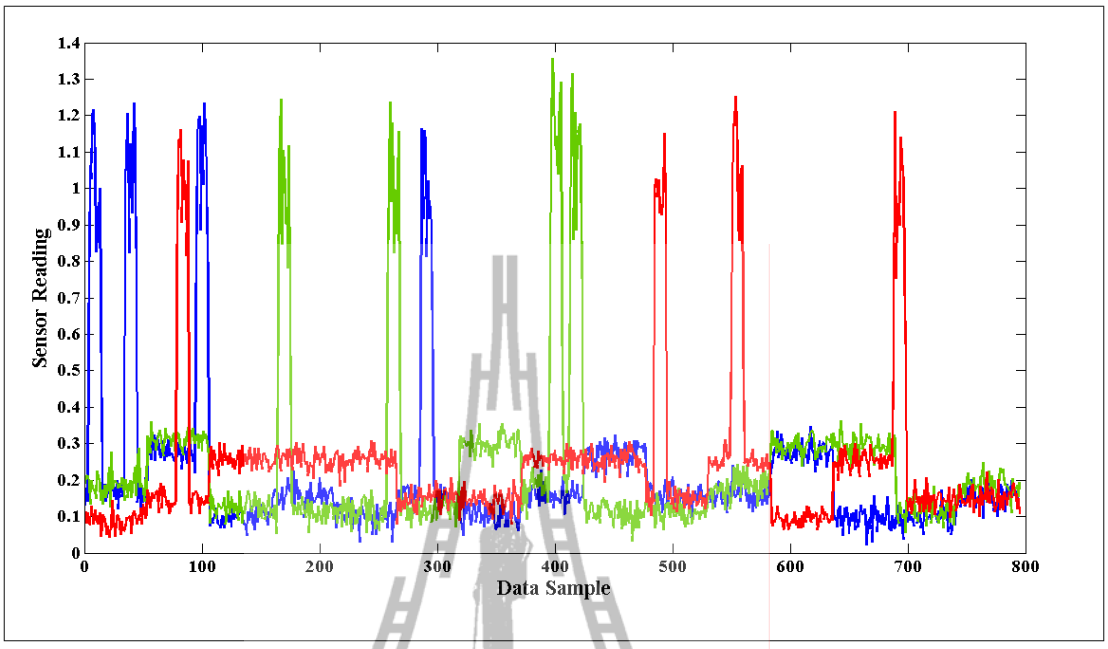


Figure B.17 DWT low-pass coefficient of 3KPI synthetic data with 20/4 faults.

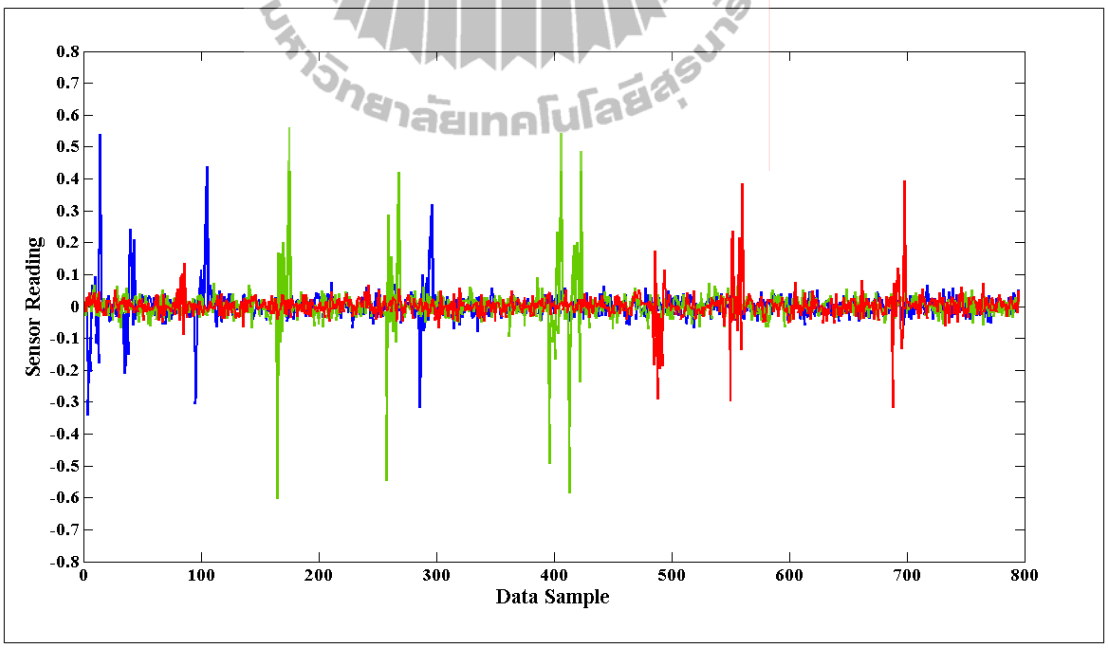


Figure B.18 DWT high-pass coefficient of 3KPI synthetic data with 20/4 faults.

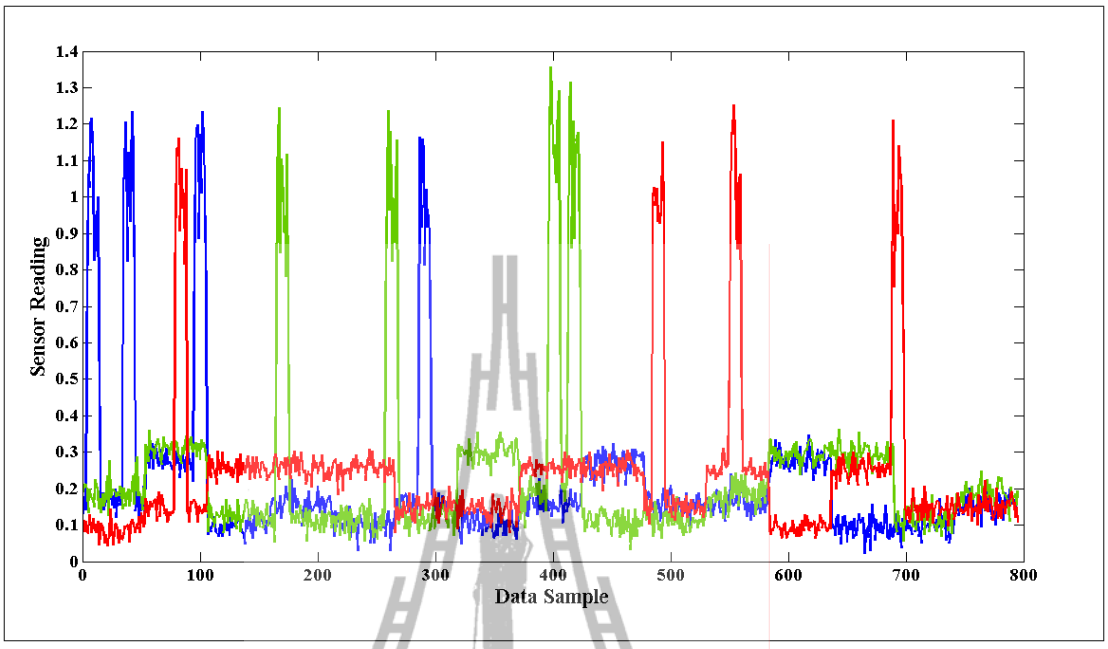


Figure B.19 LWT low-pass coefficient of 3KPI synthetic data with 20/4 faults.

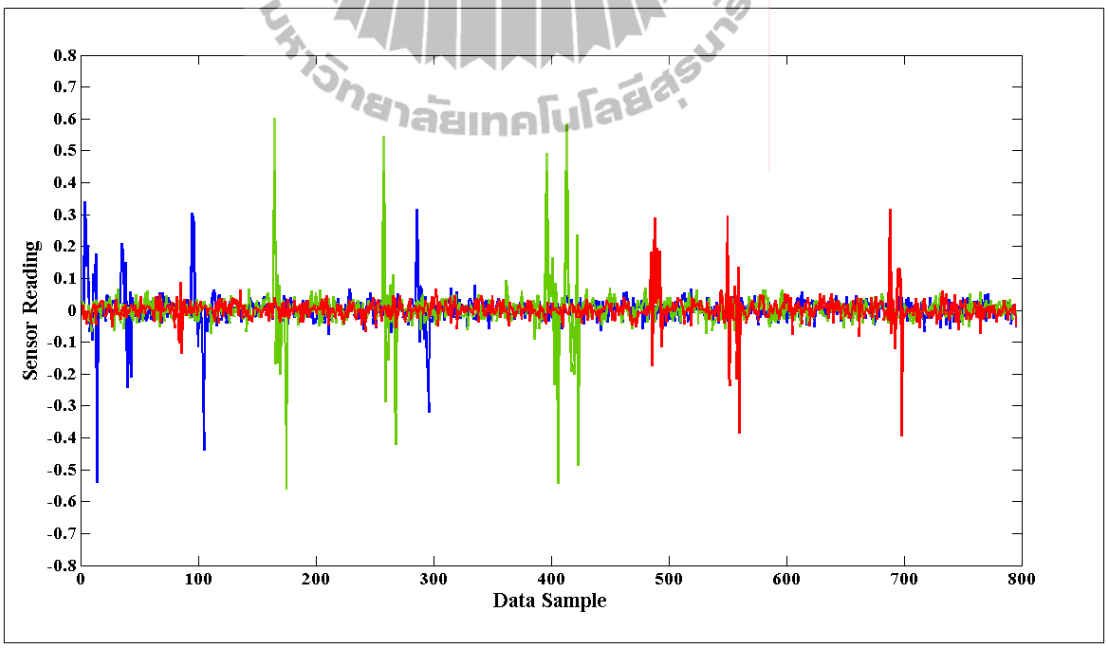


Figure B.20 LWT high-pass coefficient of 3KPI synthetic data with 20/4 faults.

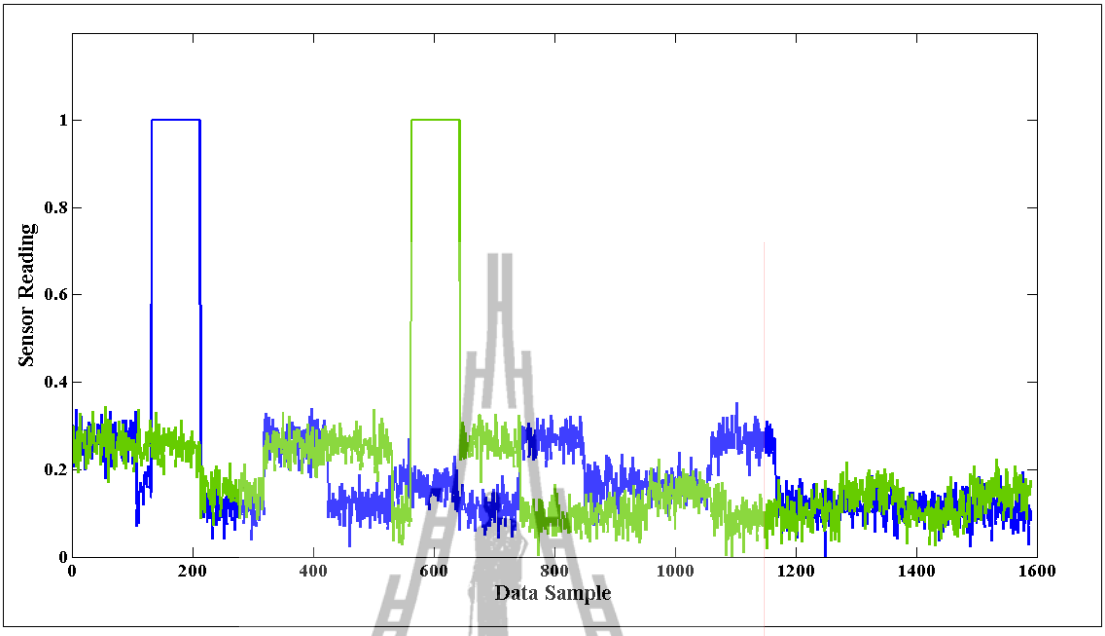


Figure B.21 2KPI Synthetic data with 80/1 fault.

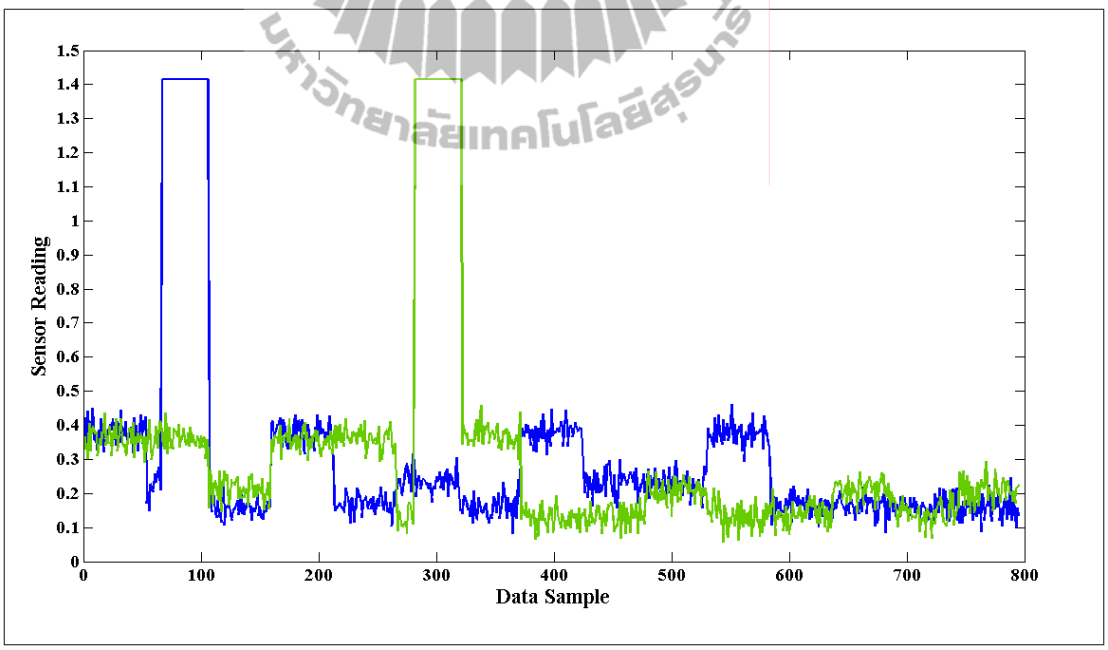


Figure B.22 DWT low-pass coefficient of 2KPI synthetic data with 80/1 faults.

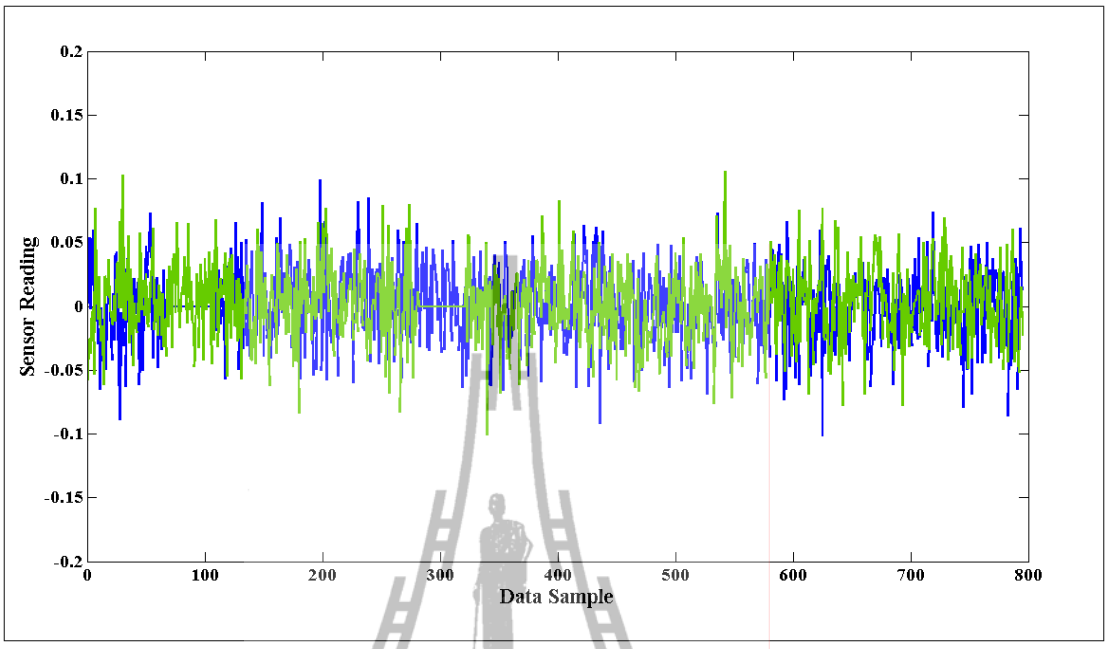


Figure B.23 DWT high-pass coefficient of 2KPI synthetic data with 80/1 faults.

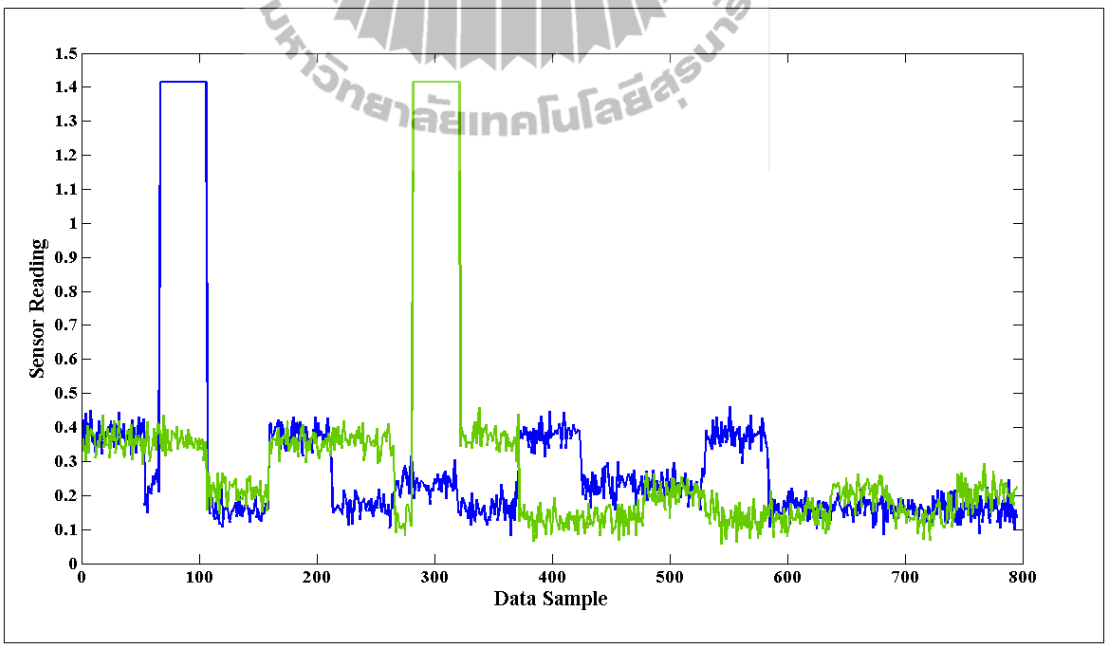


Figure B.24 LWT low-pass coefficient of 2KPI synthetic data with 80/1 faults.

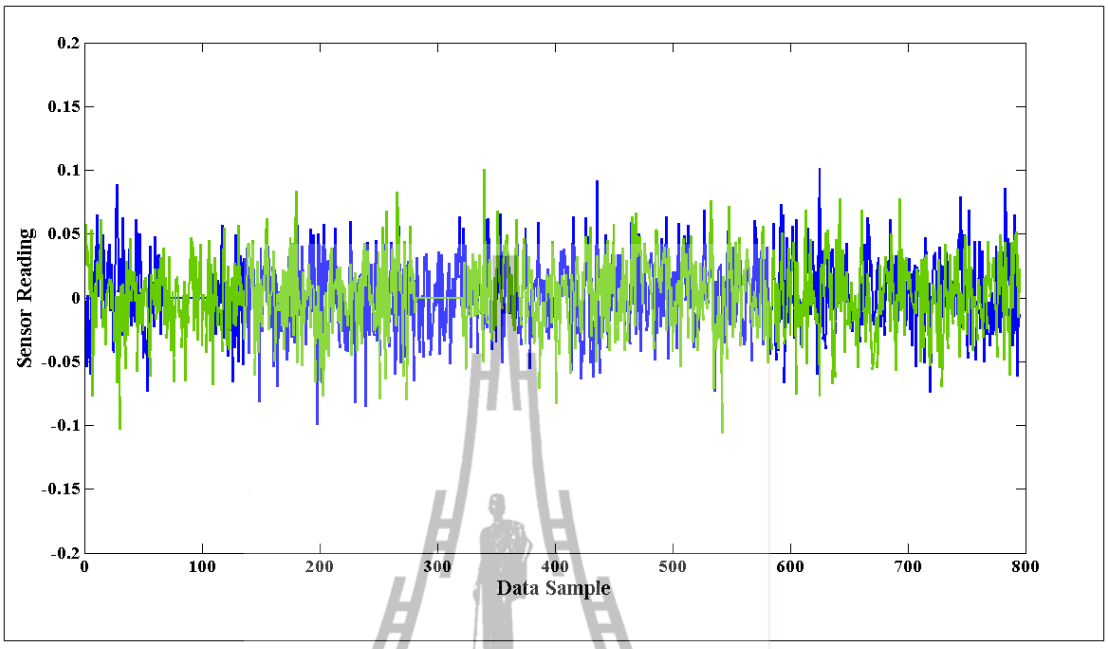


Figure B.25 LWT high-pass coefficient of 2KPI synthetic data with 80/1 faults.

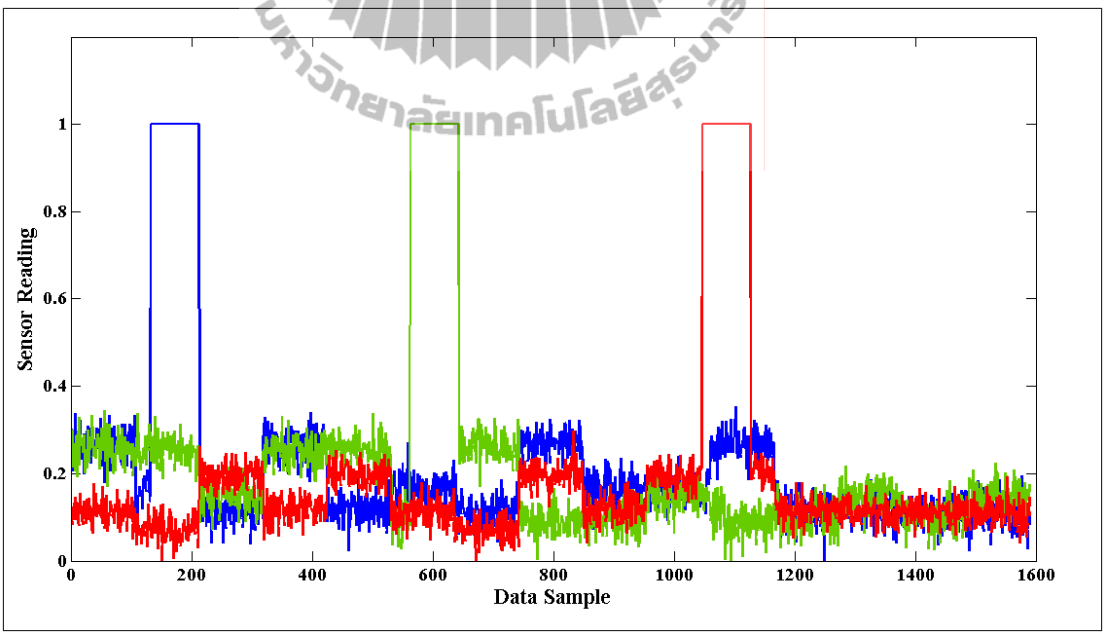


Figure B.26 3KPI Synthetic data with 80/1 faults.

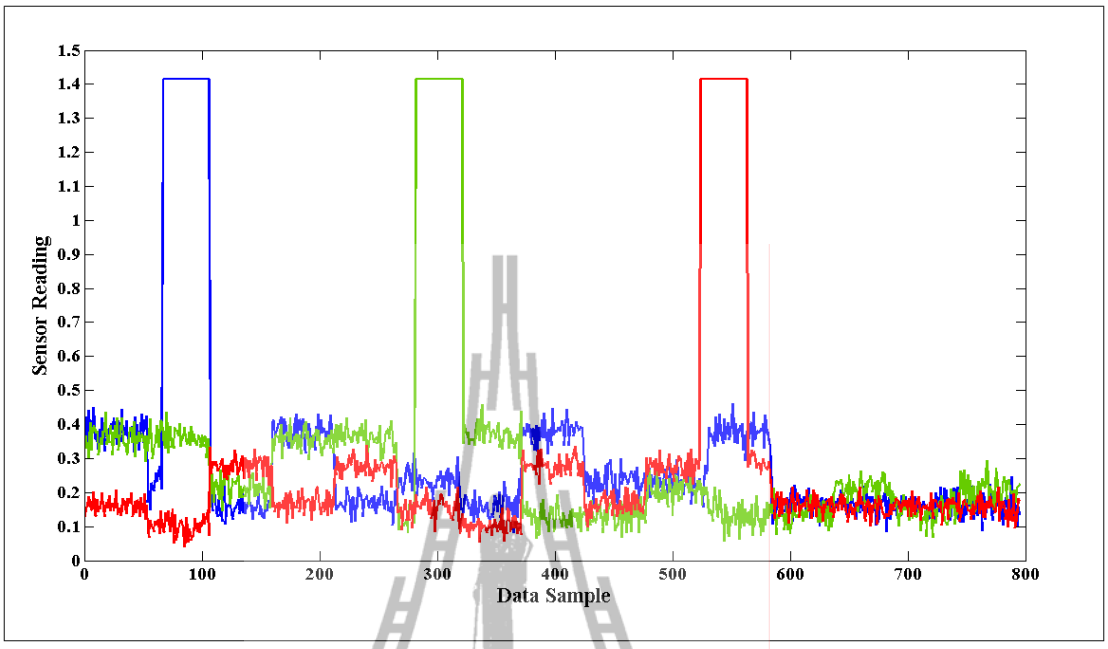


Figure B.27 DWT low-pass coefficient of 3KPI synthetic data with 80/1 faults.

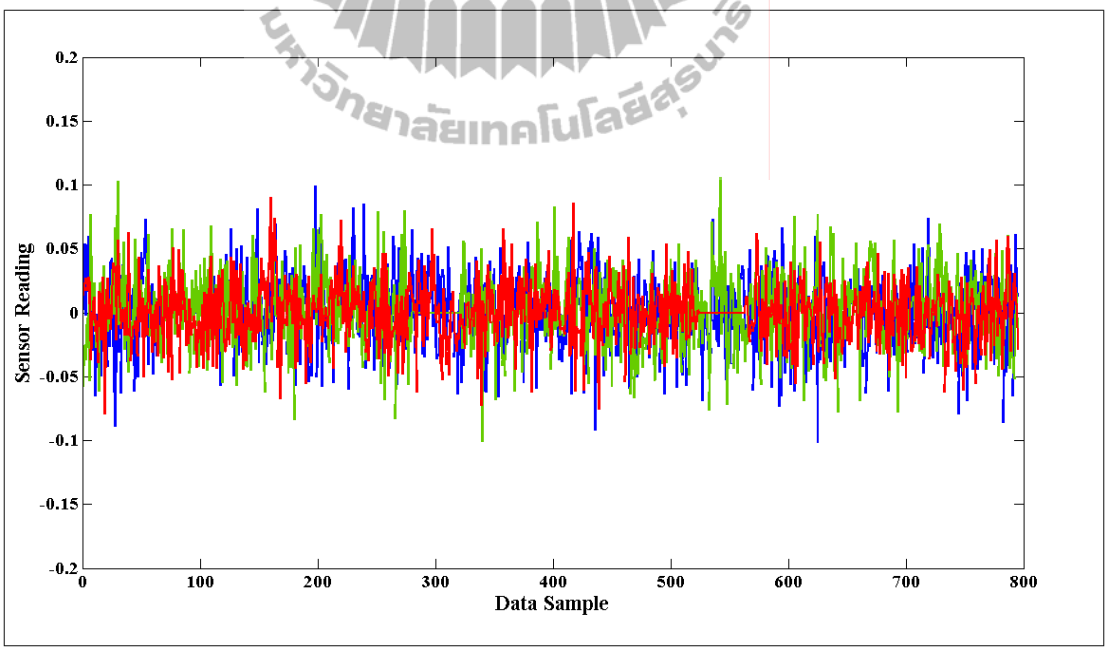


Figure B.28 DWT high-pass coefficient of 3KPI synthetic data with 80/1 faults.

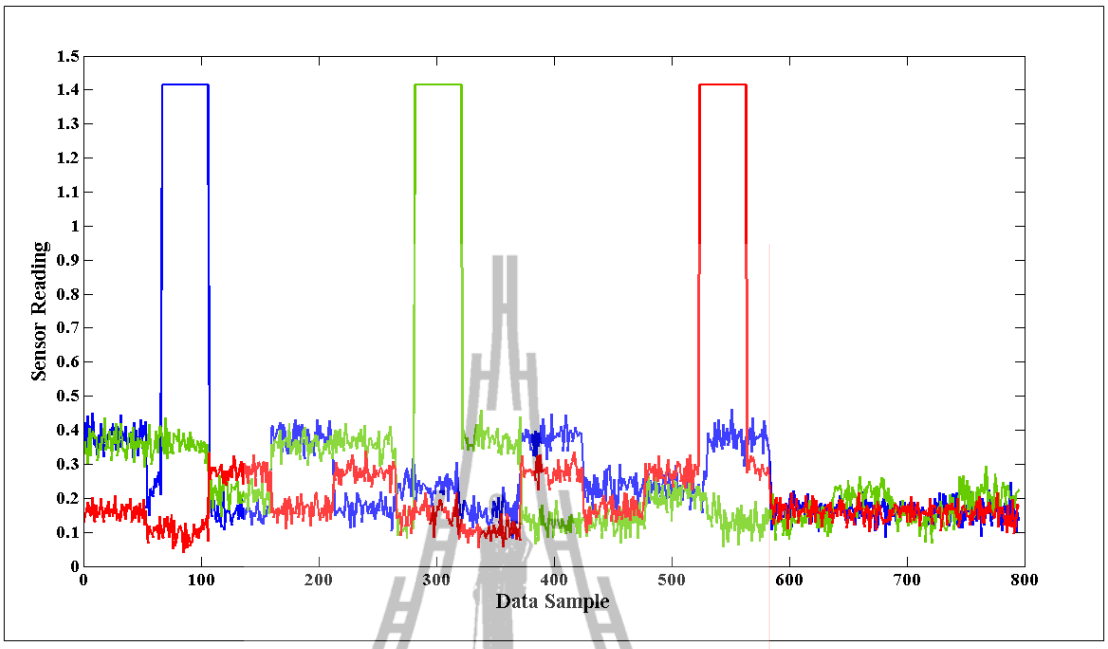


Figure B.29 LWT low-pass coefficient of 3KPI synthetic data with 80/1 faults.

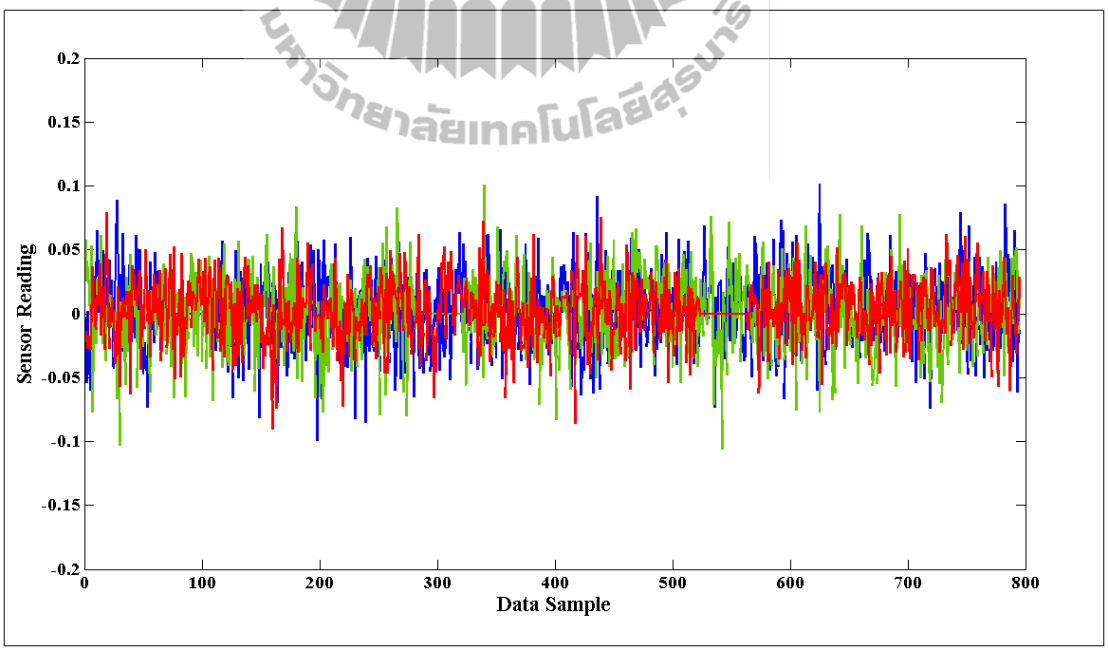


Figure B.30 LWT high-pass coefficient of 3KPI synthetic data with 80/1 faults.

2. INTEL dataset

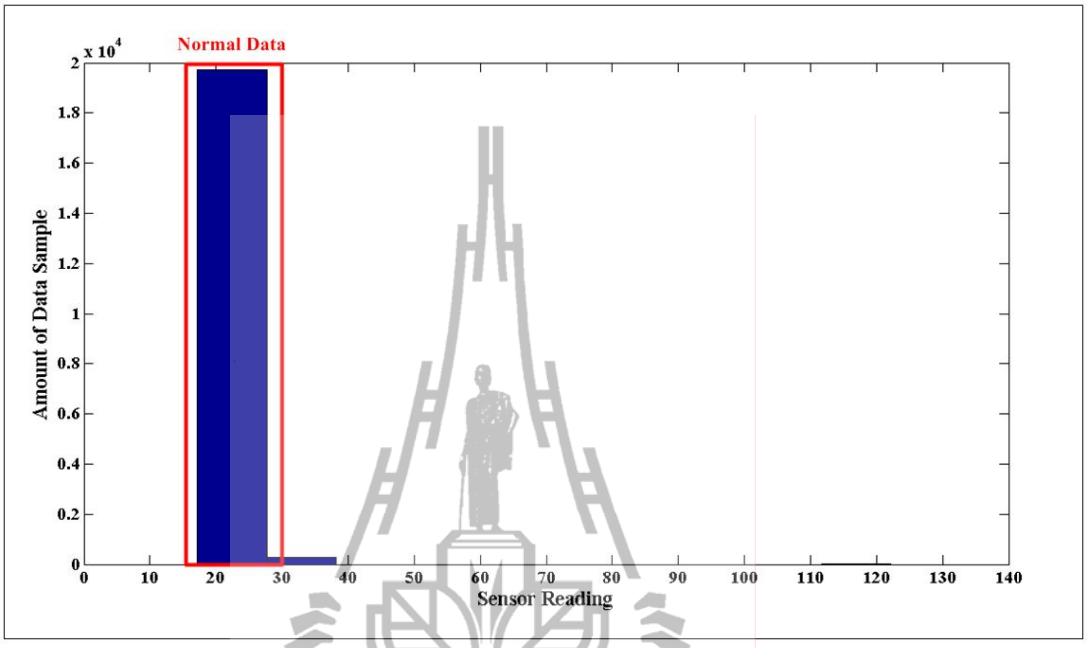


Figure B.31 Histogram of INTEL dataset (temperature reading).

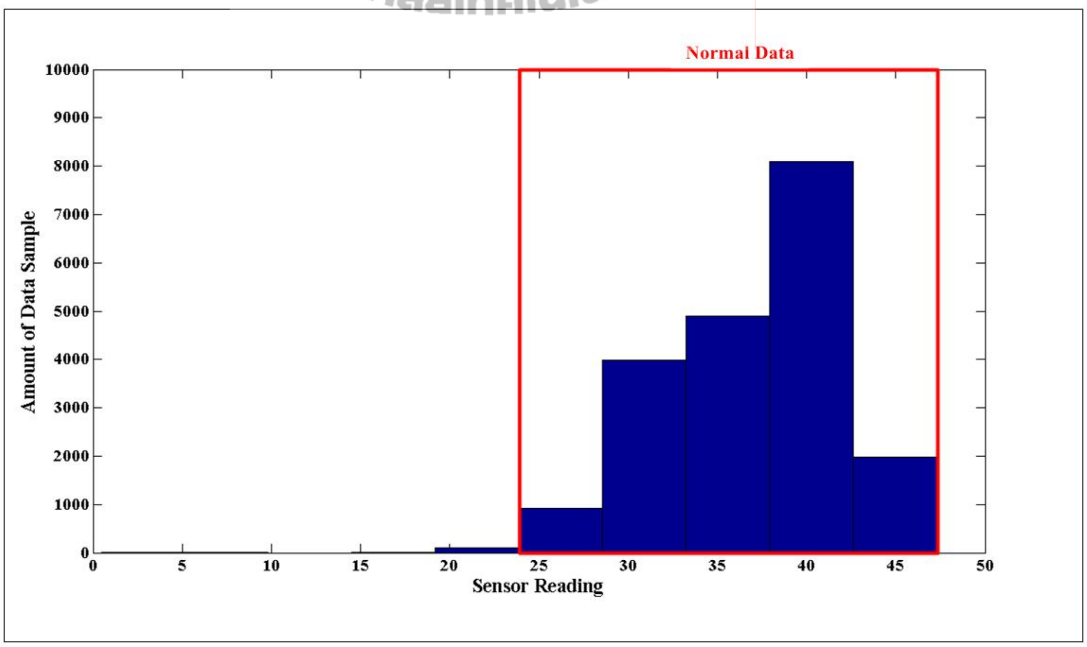


Figure B.32 Histogram of INTEL dataset (humidity reading).

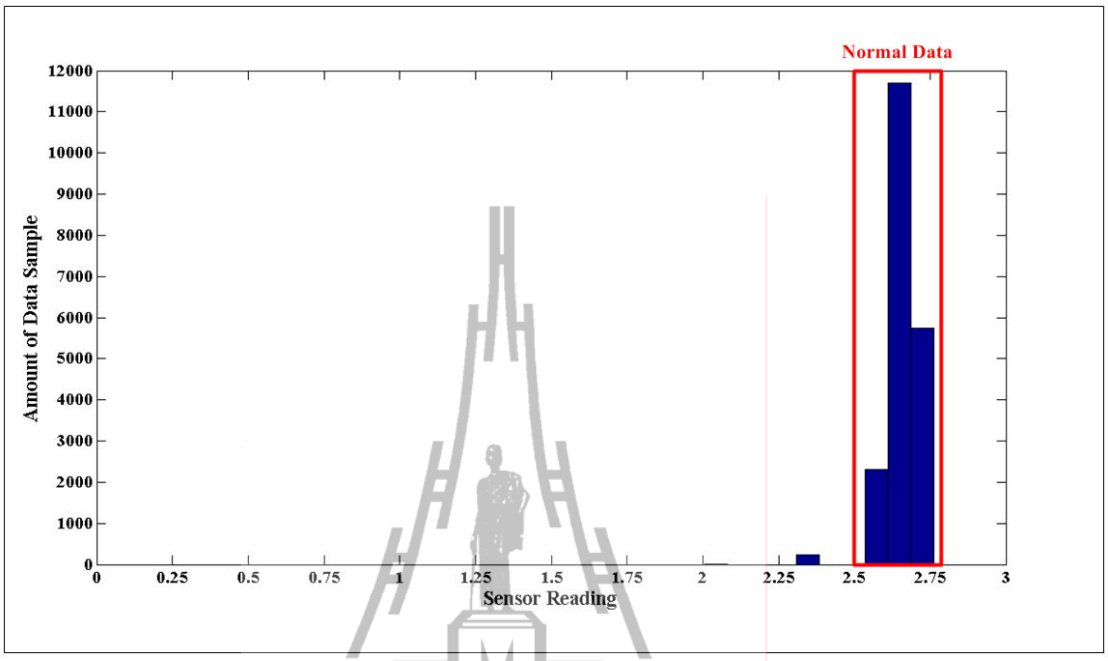


Figure B.33 Histogram of INTEL dataset (voltage reading).

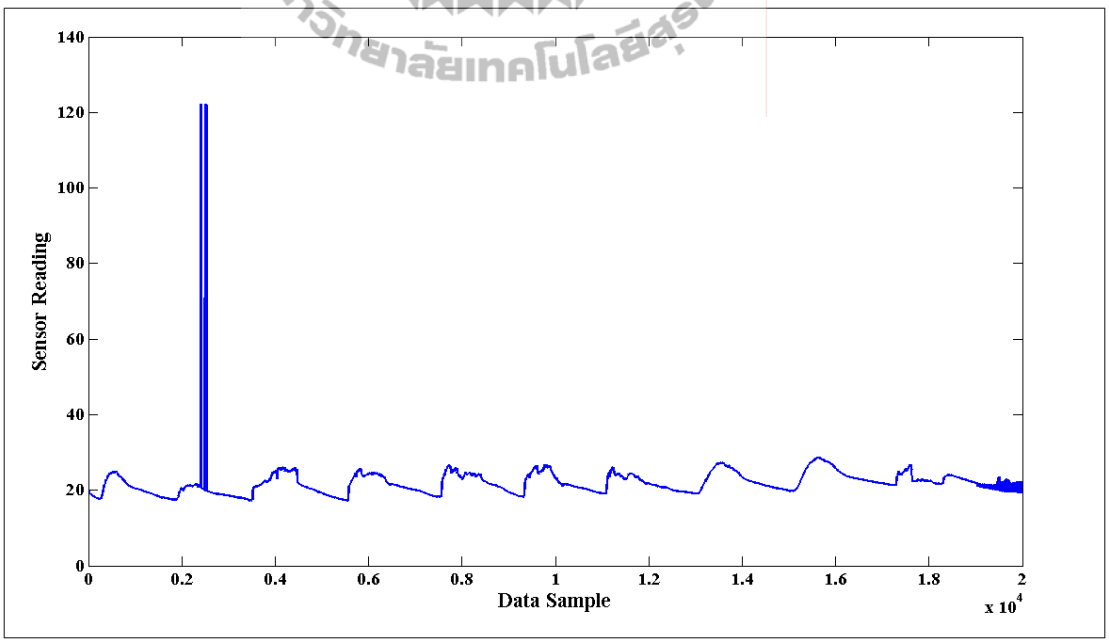


Figure B.34 INTEL dataset (temperature reading).

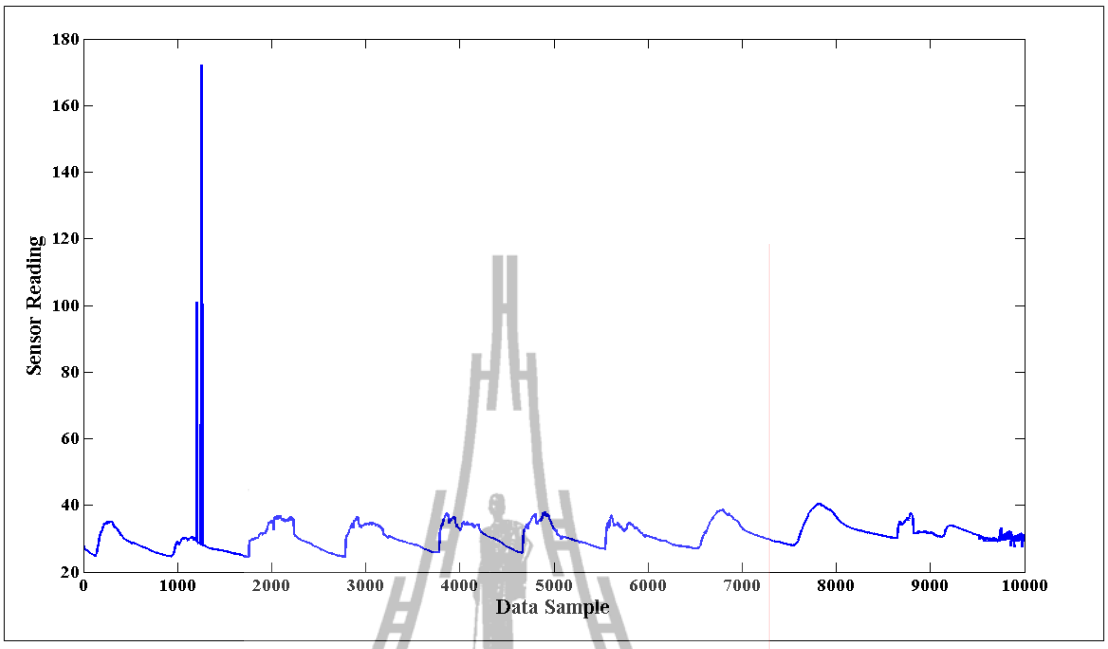


Figure B.35 DWT low-pass coefficient of INTEL dataset (temperature reading).

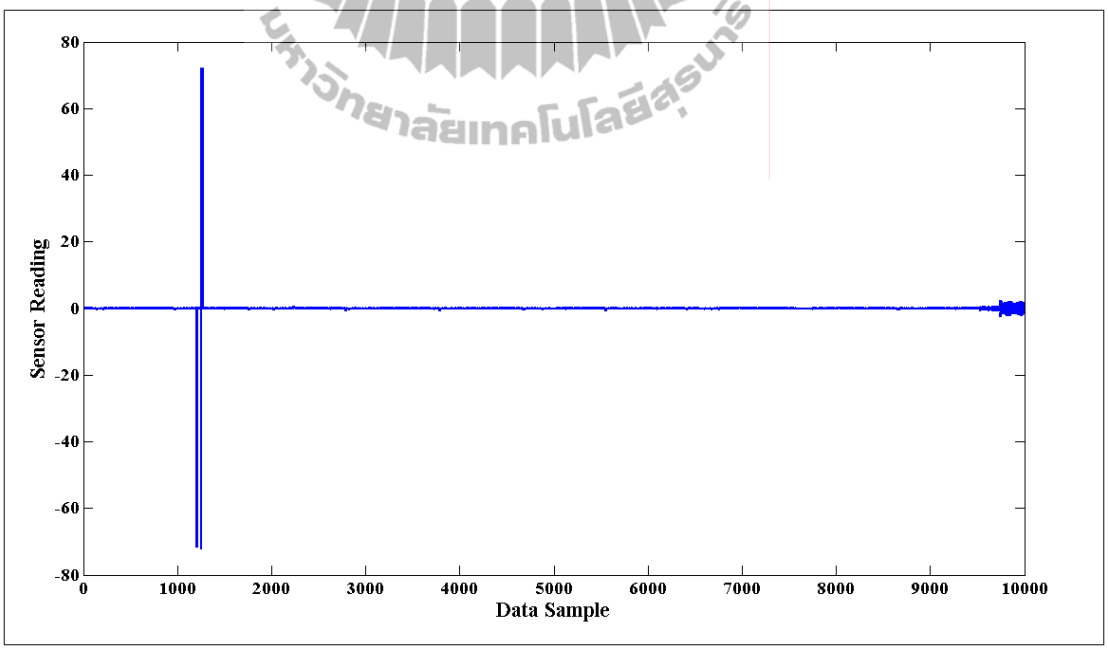


Figure B.36 DWT high-pass coefficient of INTEL dataset (temperature reading).

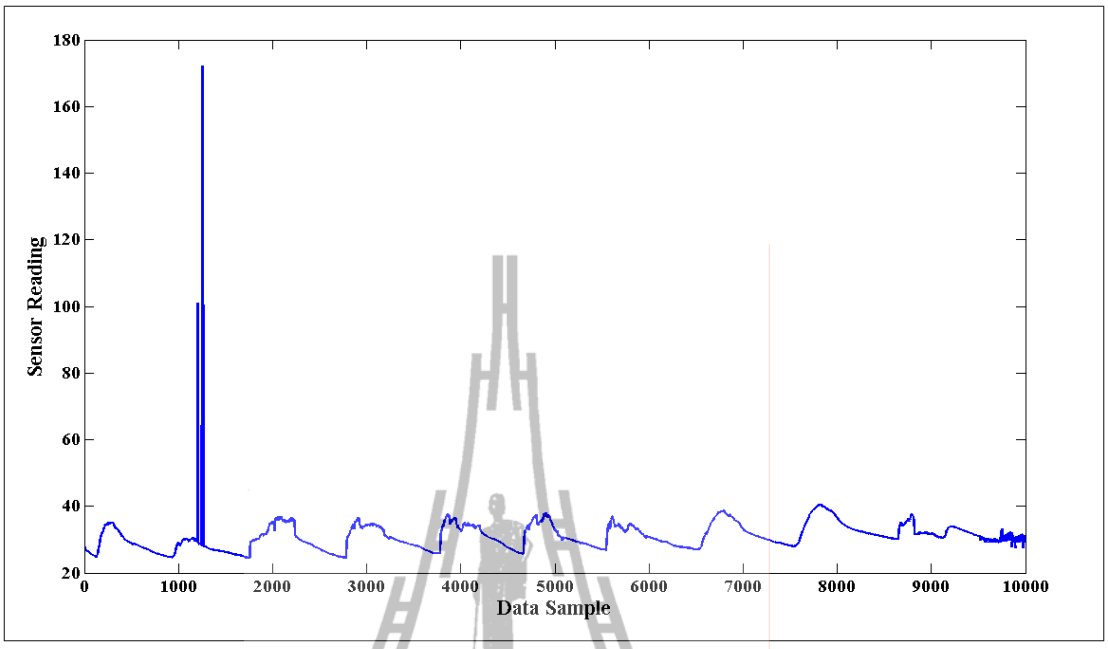


Figure B.37 LWT low-pass coefficient of INTEL dataset (temperature reading).

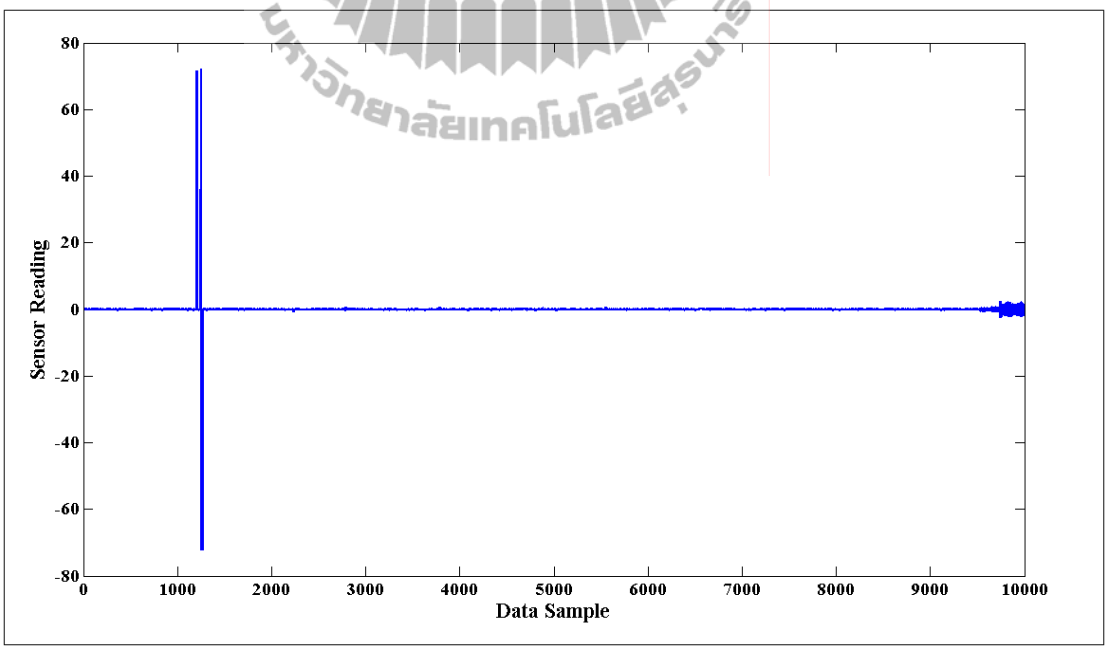


Figure B.38 LWT high-pass coefficient of INTEL dataset (temperature reading).

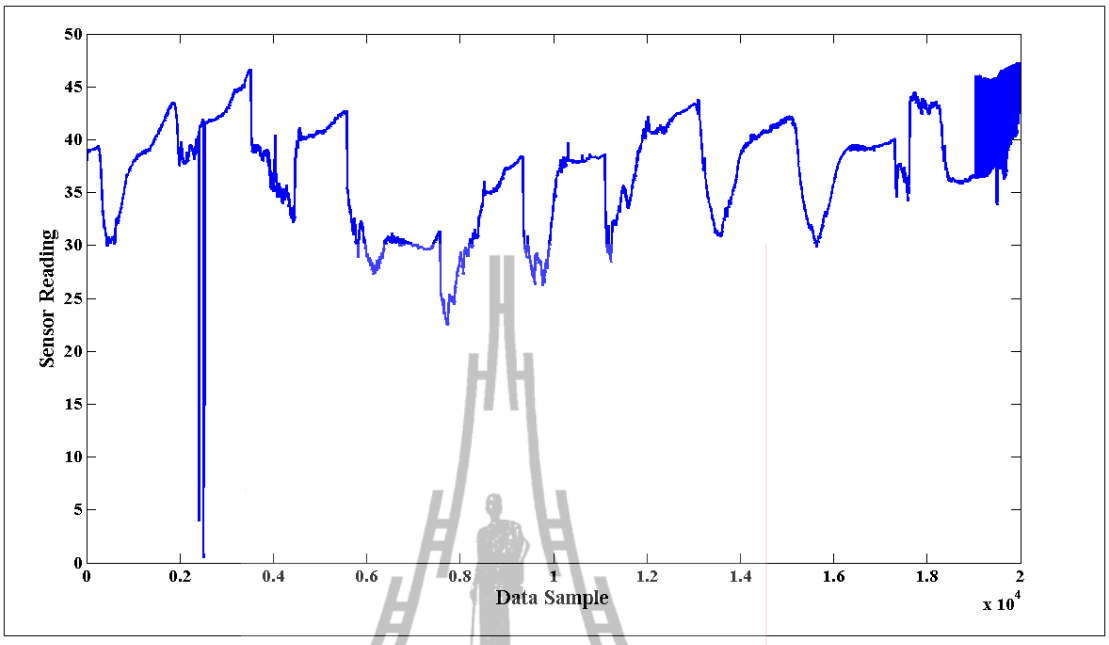


Figure B.39 INTEL dataset (humidity reading).

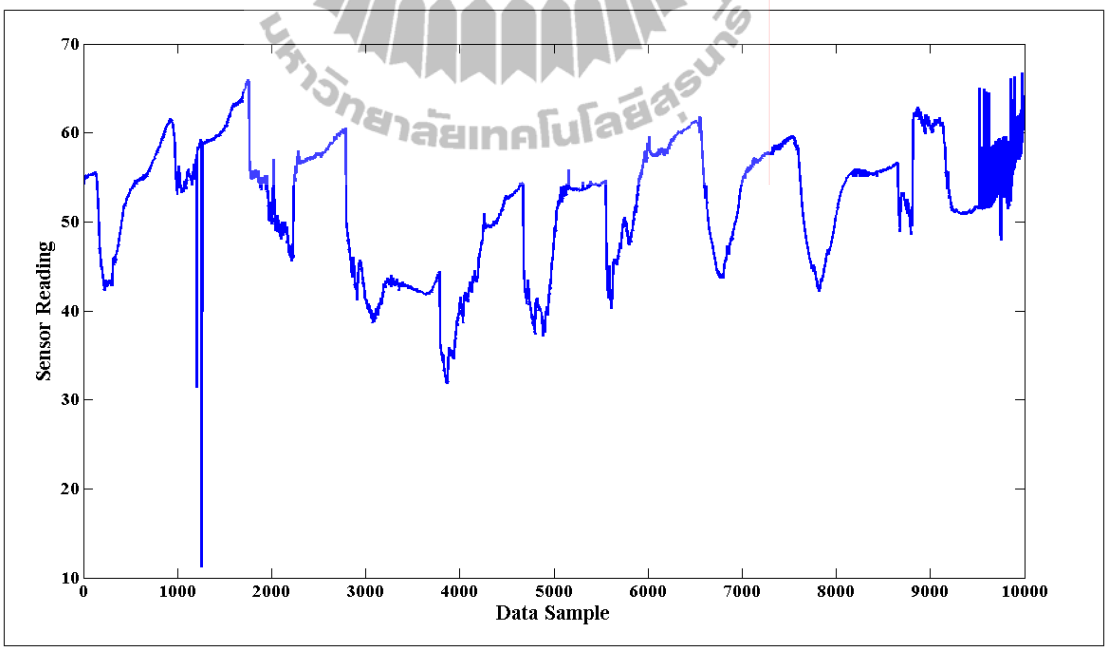


Figure B.40 DWT low-pass coefficient of INTEL dataset (humidity reading).

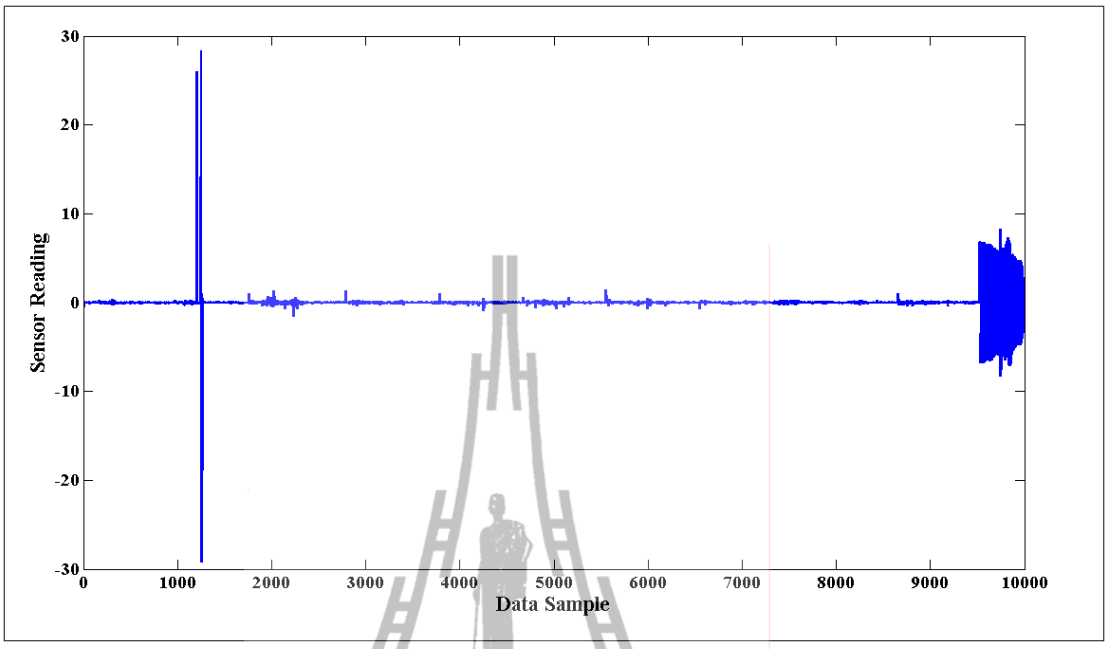


Figure B.41 DWT high-pass coefficient of INTEL dataset (humidity reading).

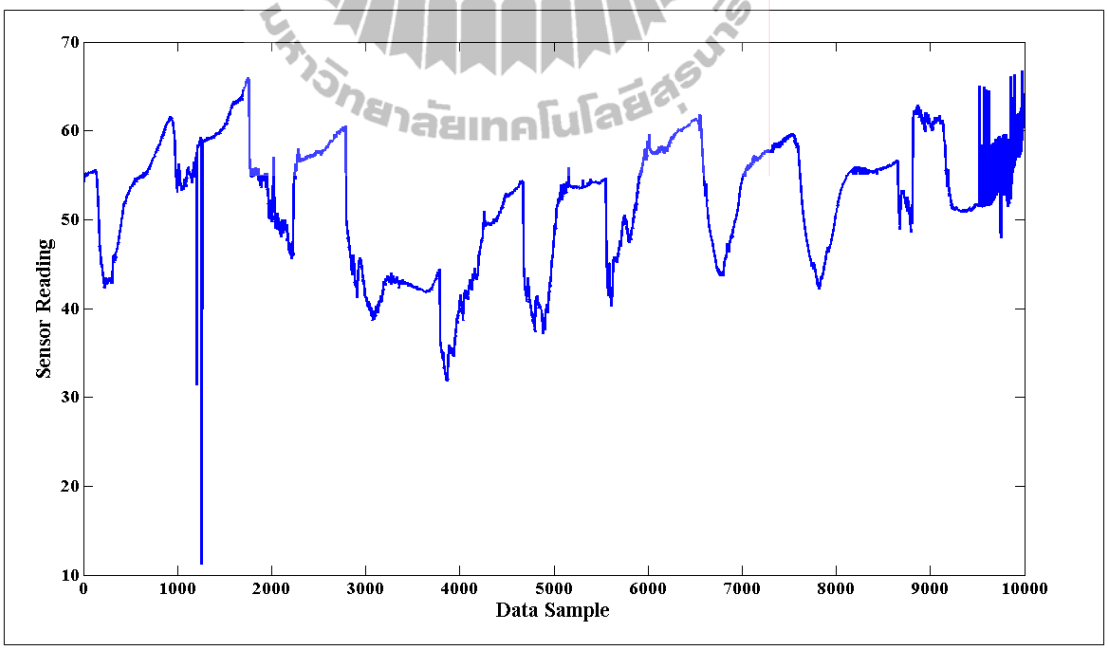


Figure B.42 LWT low-pass coefficient of INTEL dataset (humidity reading).

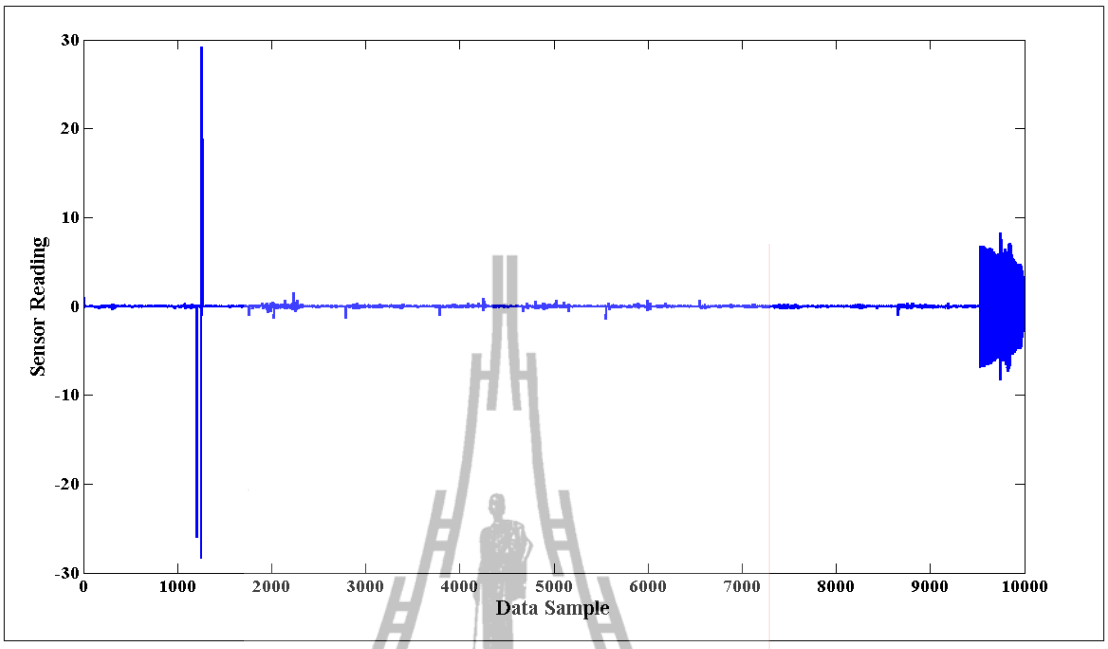


Figure B.43 LWT high-pass coefficient of INTEL dataset (humidity reading).

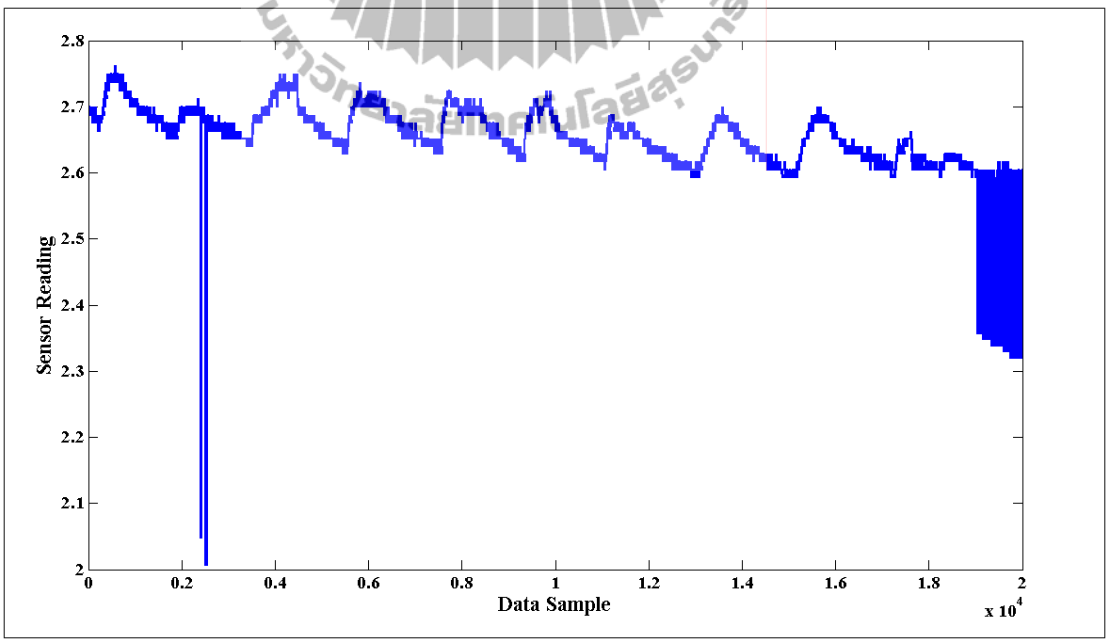


Figure B.44 INTEL dataset (voltage reading).

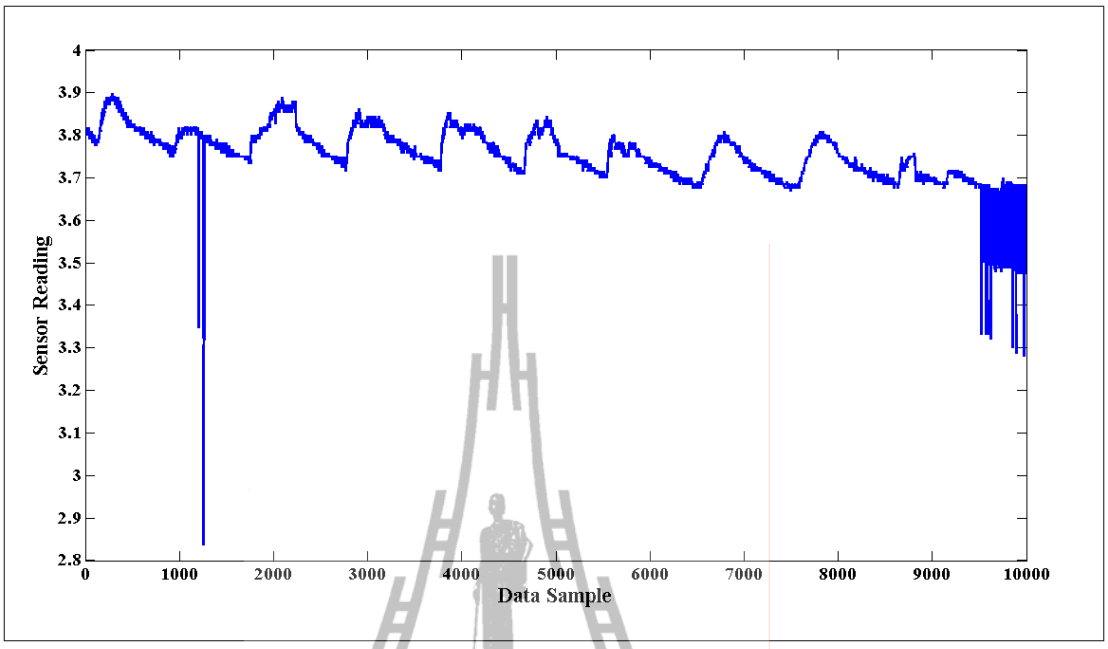


Figure B.45 DWT low-pass coefficient of INTEL dataset (voltage reading).

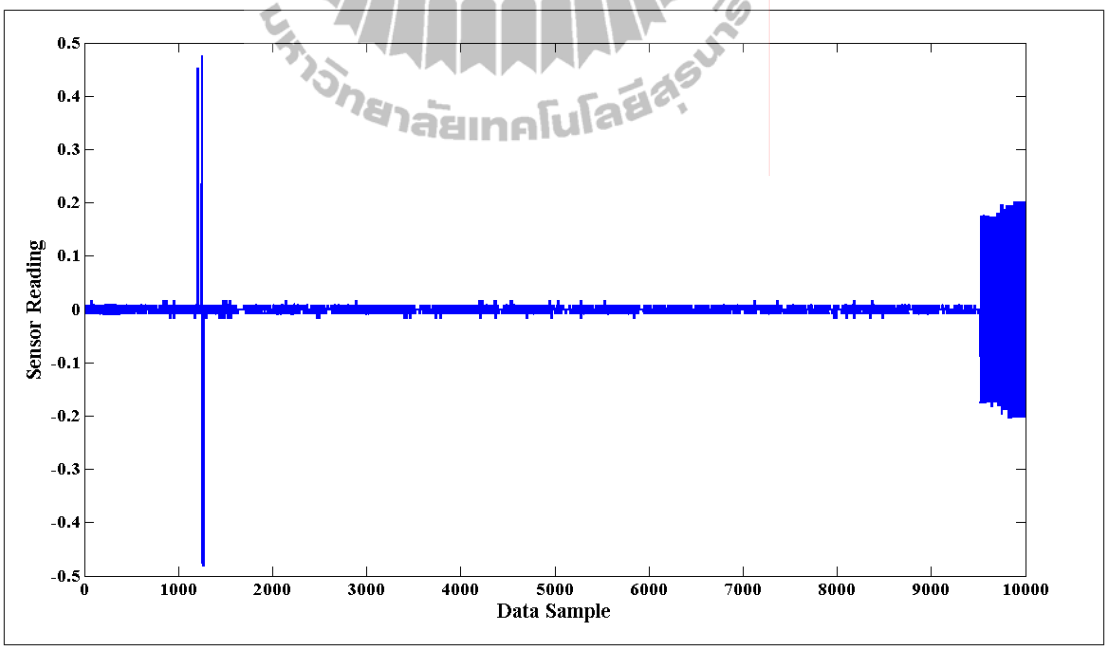


Figure B.46 DWT high-pass coefficient of INTEL dataset (voltage reading).

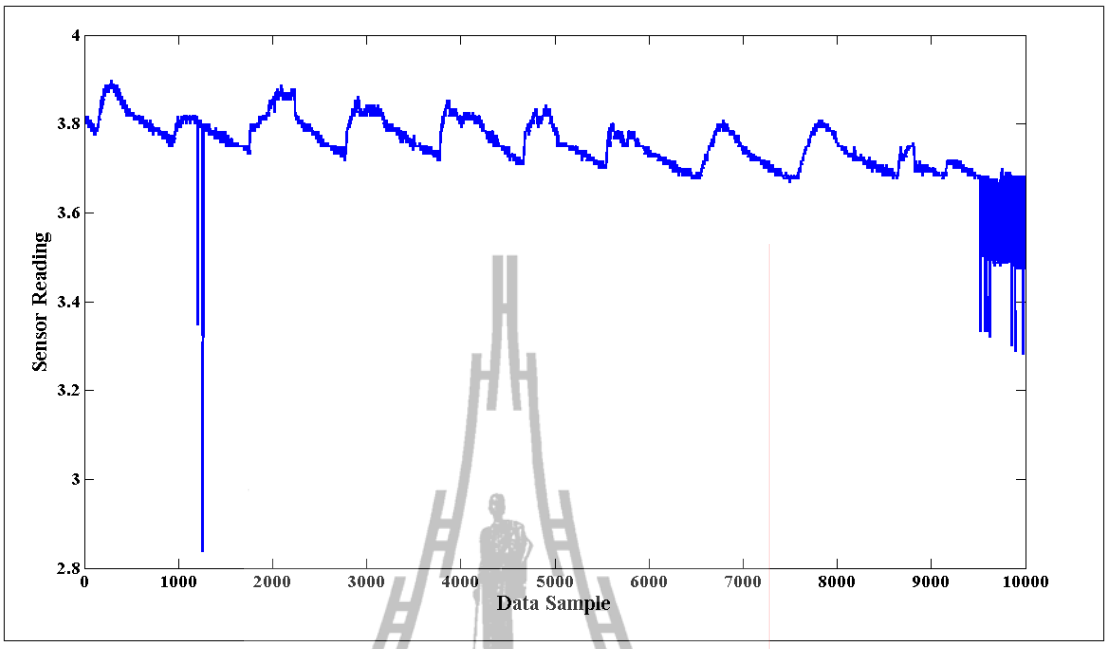


Figure B.47 LWT low-pass coefficient of INTEL dataset (voltage reading).

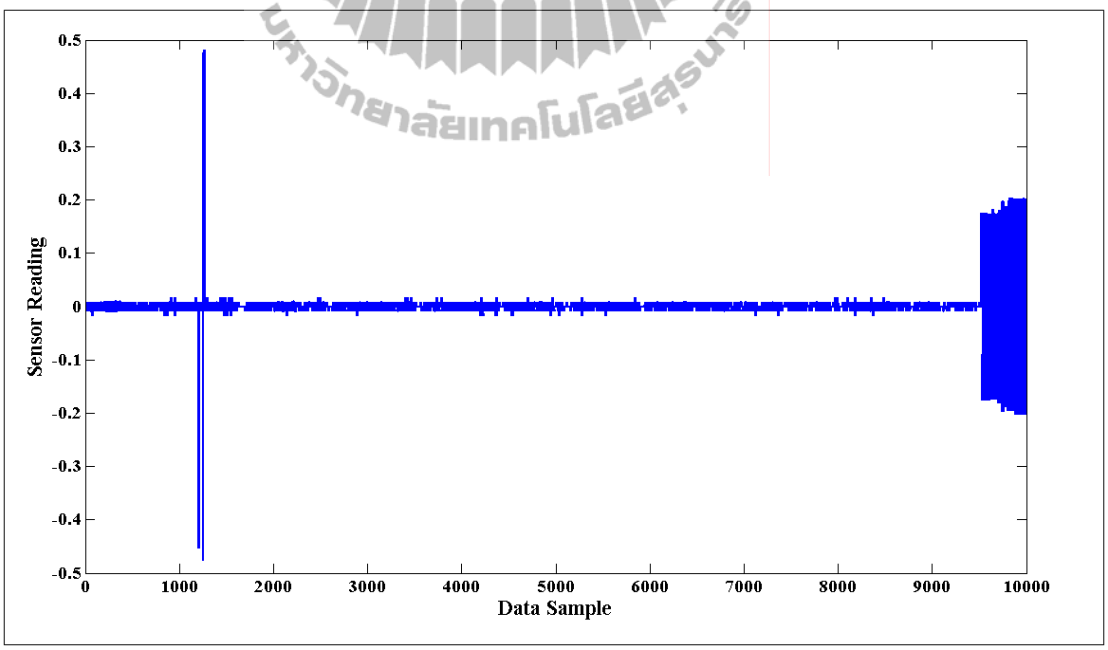


Figure B.48 LWT high-pass coefficient of INTEL dataset (voltage reading).

3. SensorScope pdg2008-metro-1 dataset (pdg2008)

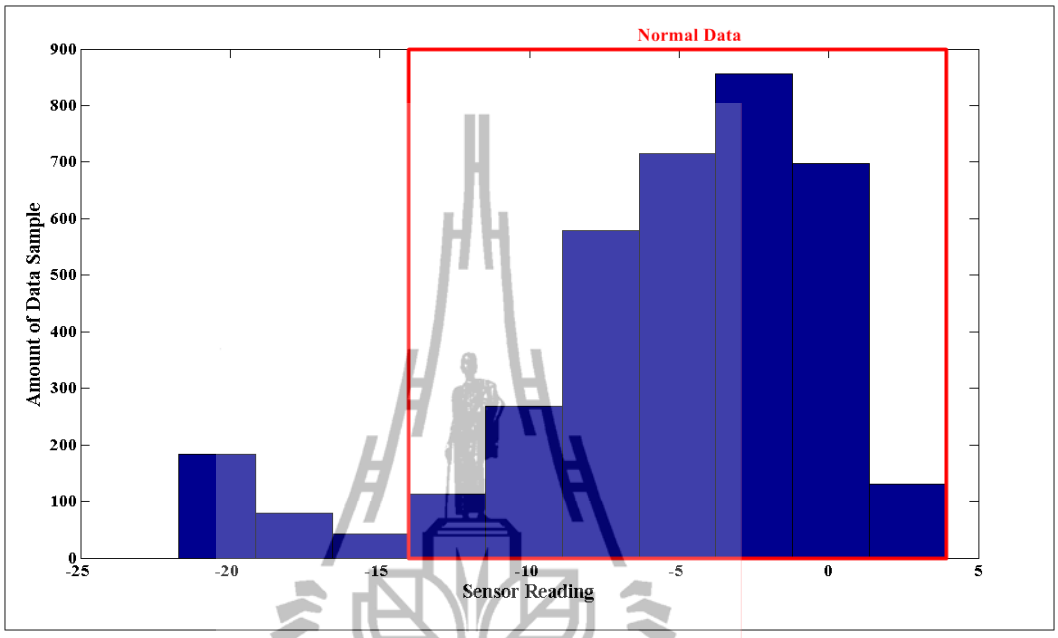


Figure B.49 Histogram of pdg2008 dataset (surface temperature reading).

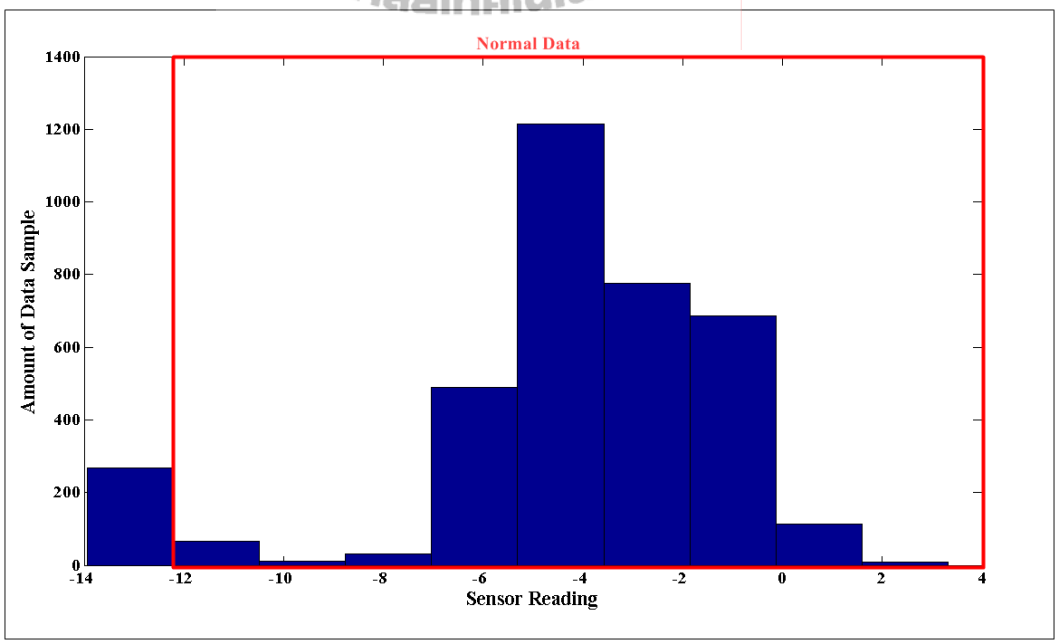


Figure B.50 Histogram of pdg2008 dataset (ambient temperature reading).

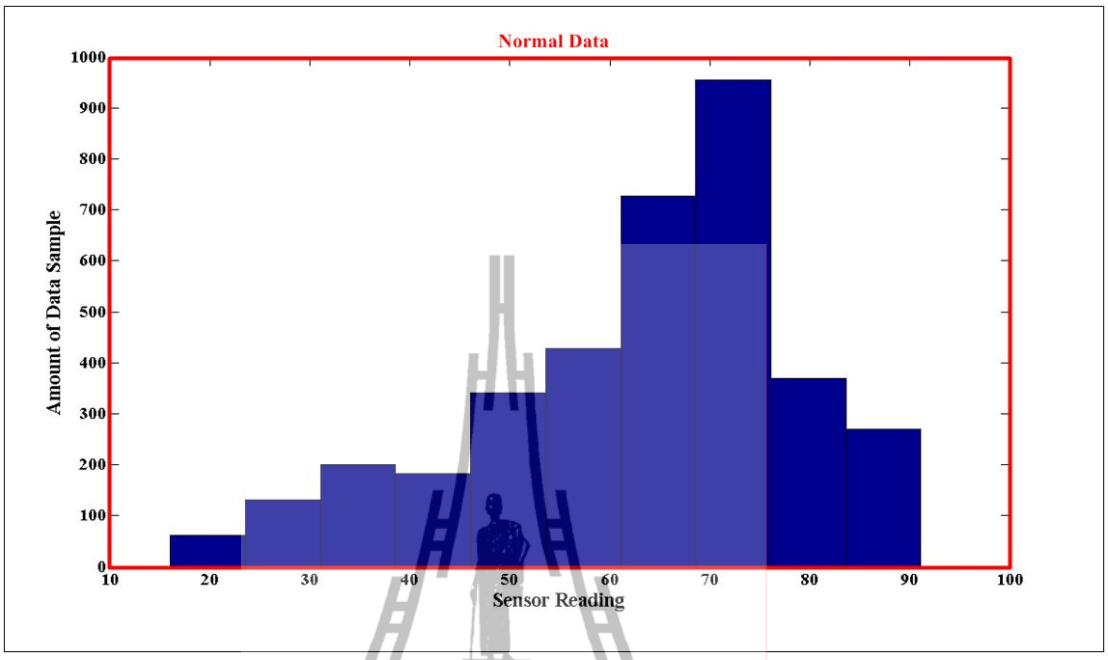


Figure B.51 Histogram of pdg2008 dataset (solar radiation reading).

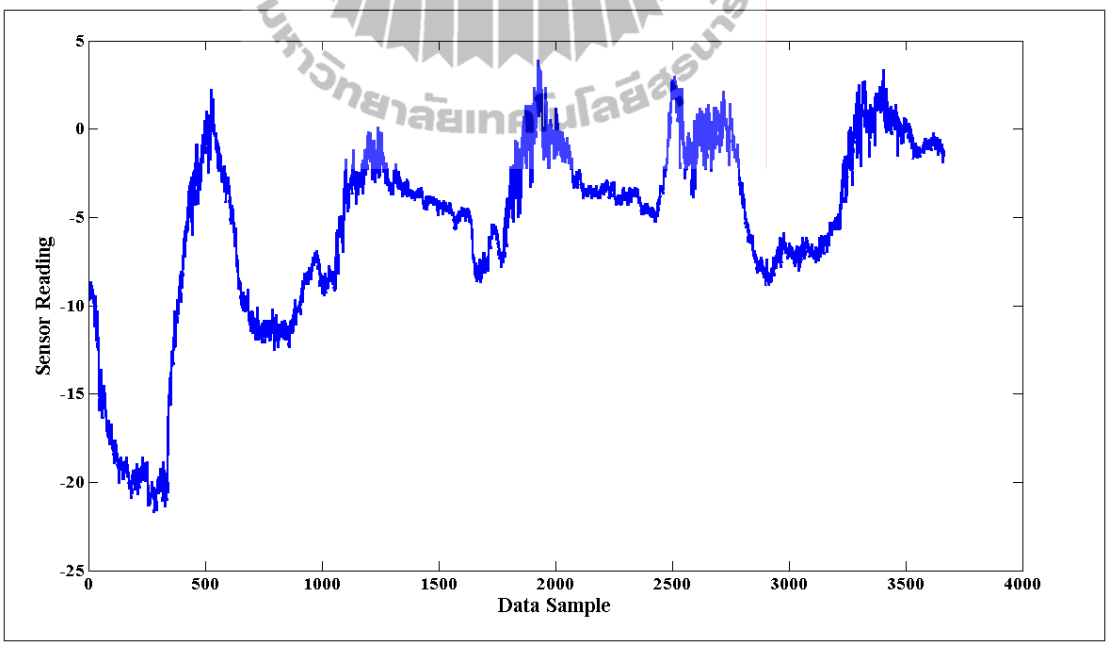


Figure B.52 pdg2008 dataset (surface temperature reading).

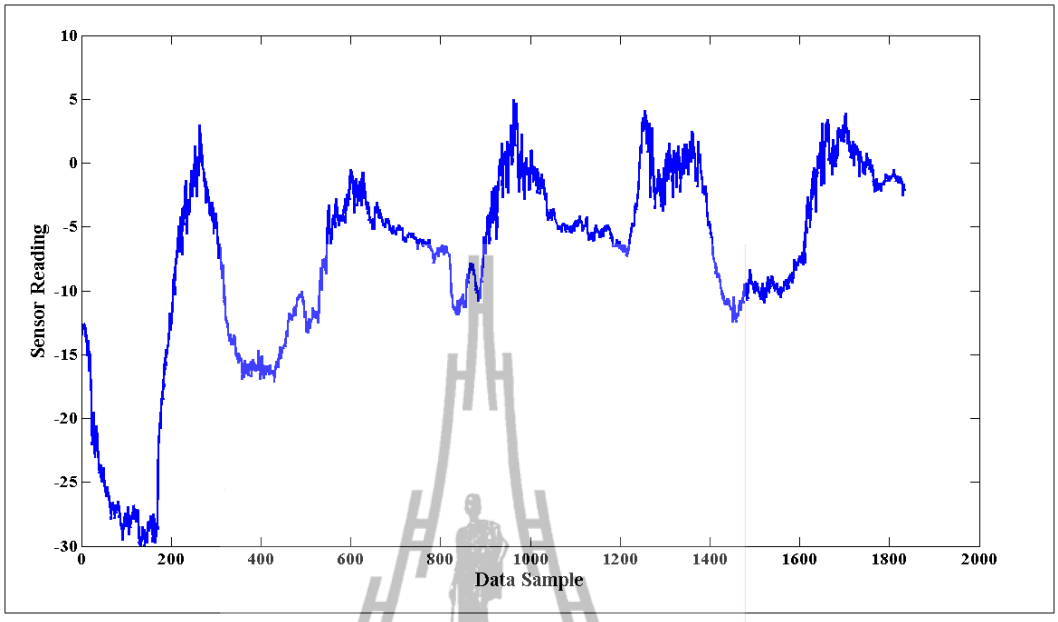


Figure B.53 DWT low-pass coefficient of pdg2008 dataset (surface temperature reading).

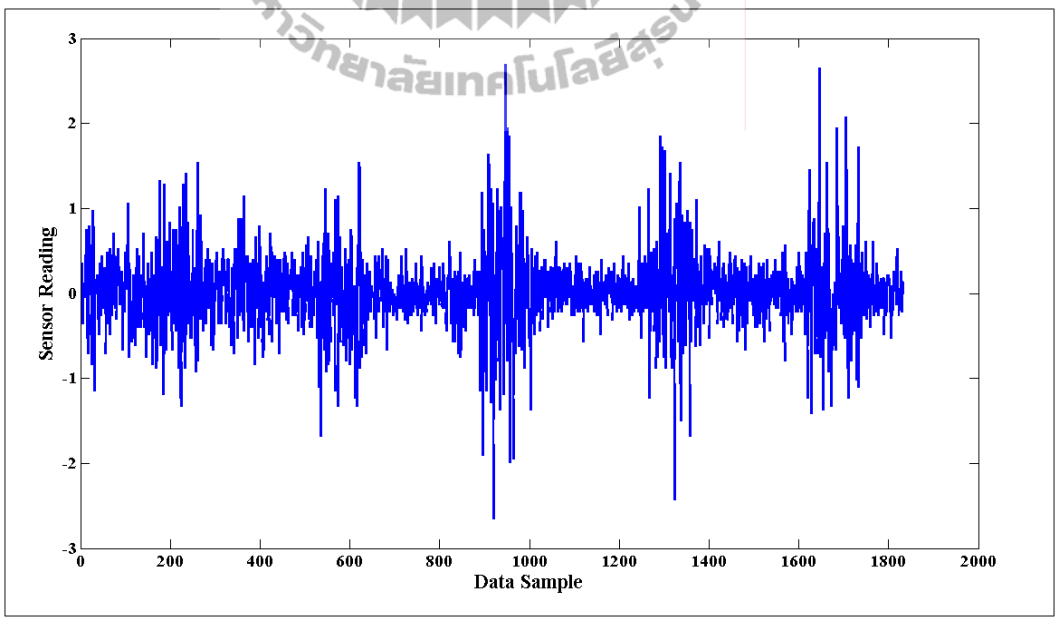


Figure B.54 DWT high-pass coefficient of pdg2008 dataset (surface temperature reading).

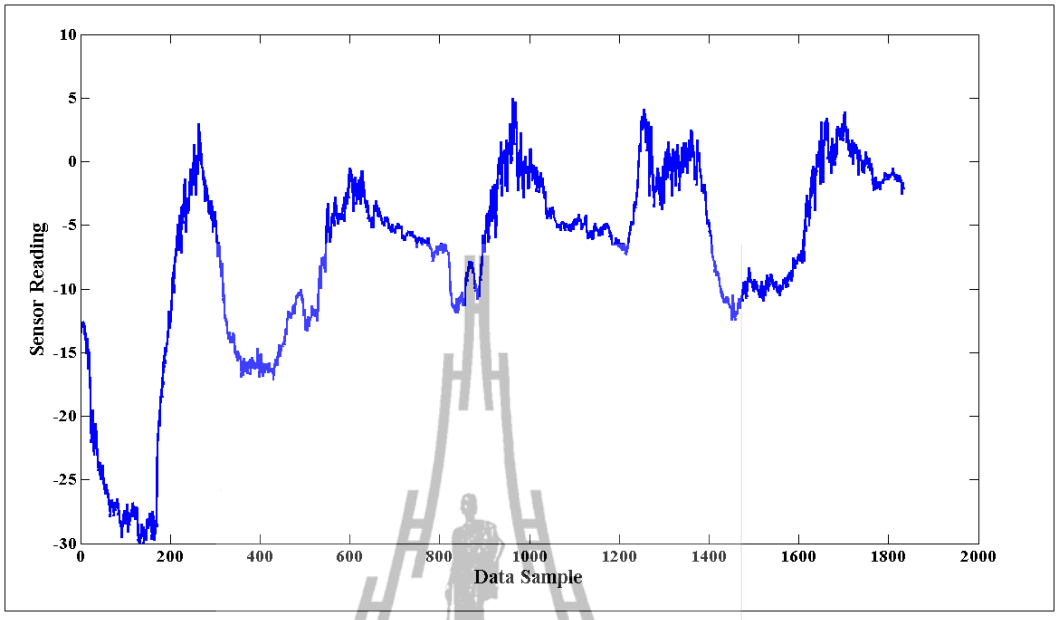


Figure B.55 LWT low-pass coefficient of pdg2008 dataset (surface temperature reading).

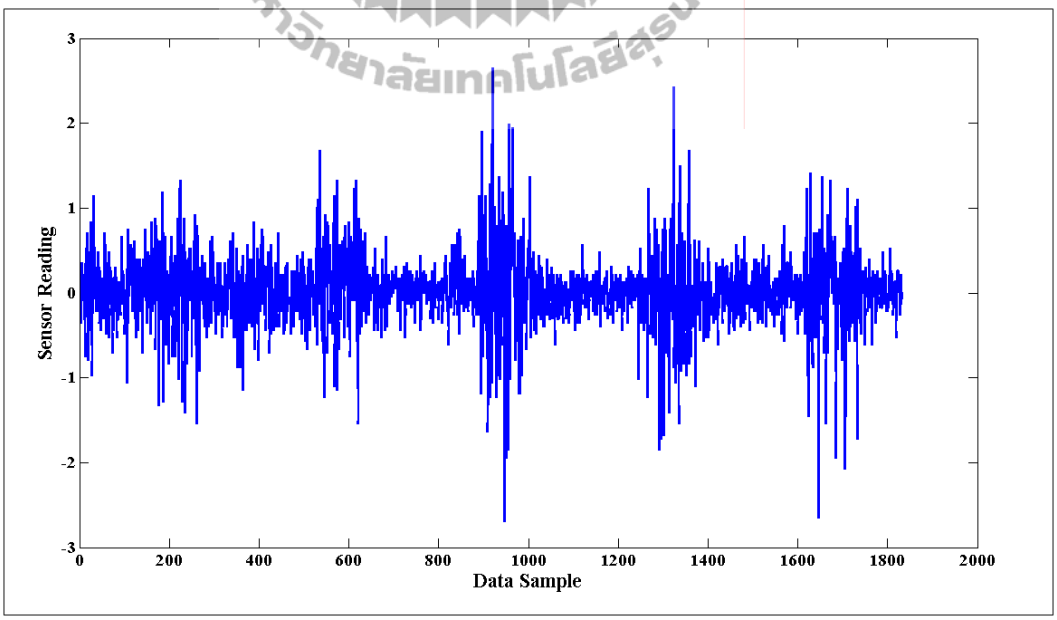


Figure B.56 LWT high-pass coefficient of pdg2008 dataset (surface temperature reading).

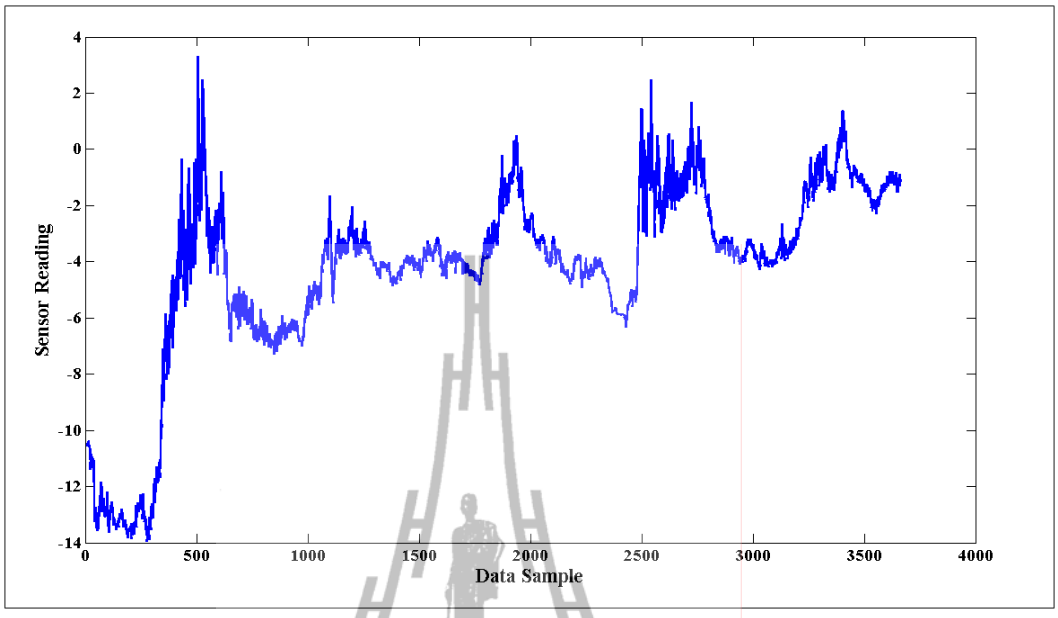


Figure B.57 pdg2008 dataset (ambient temperature reading).

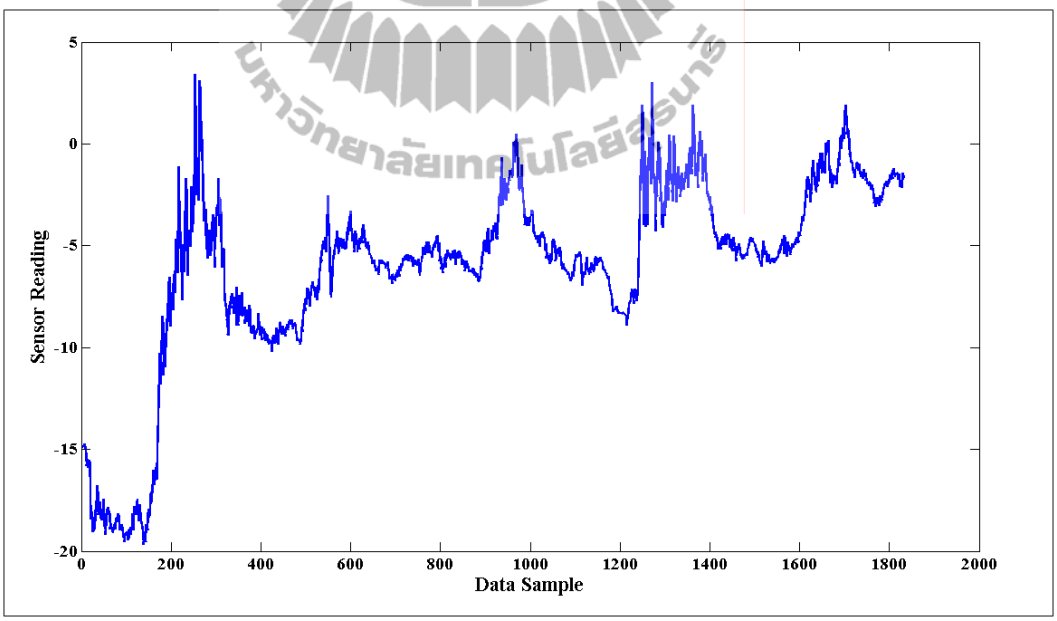


Figure B.58 DWT low-pass coefficient of pdg2008 dataset

(ambient temperature reading).

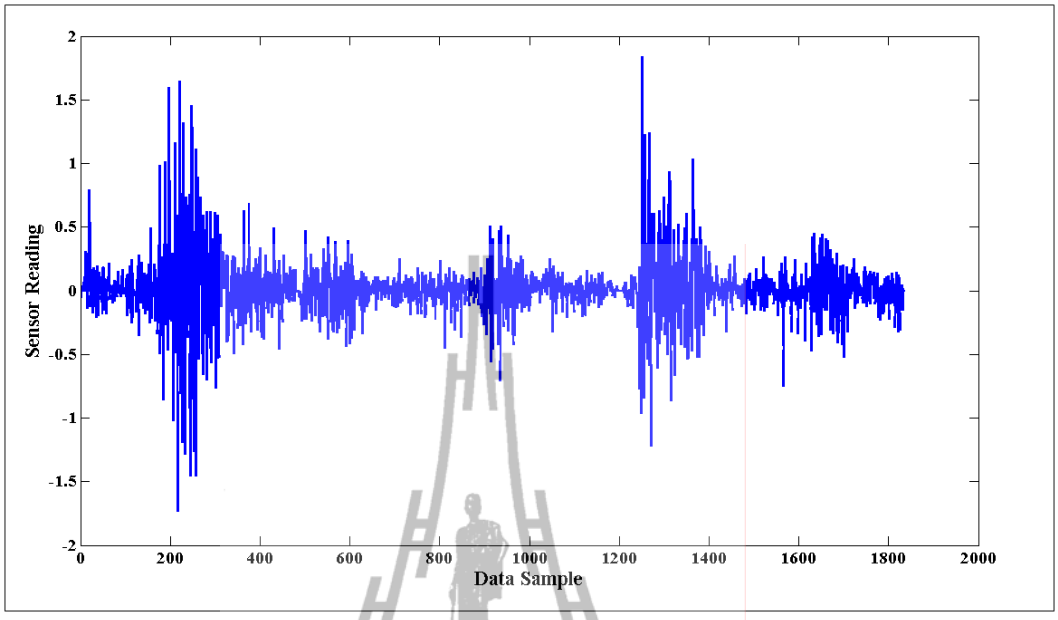


Figure B.59 DWT high-pass coefficient of pdg2008 dataset (ambient temperature reading).

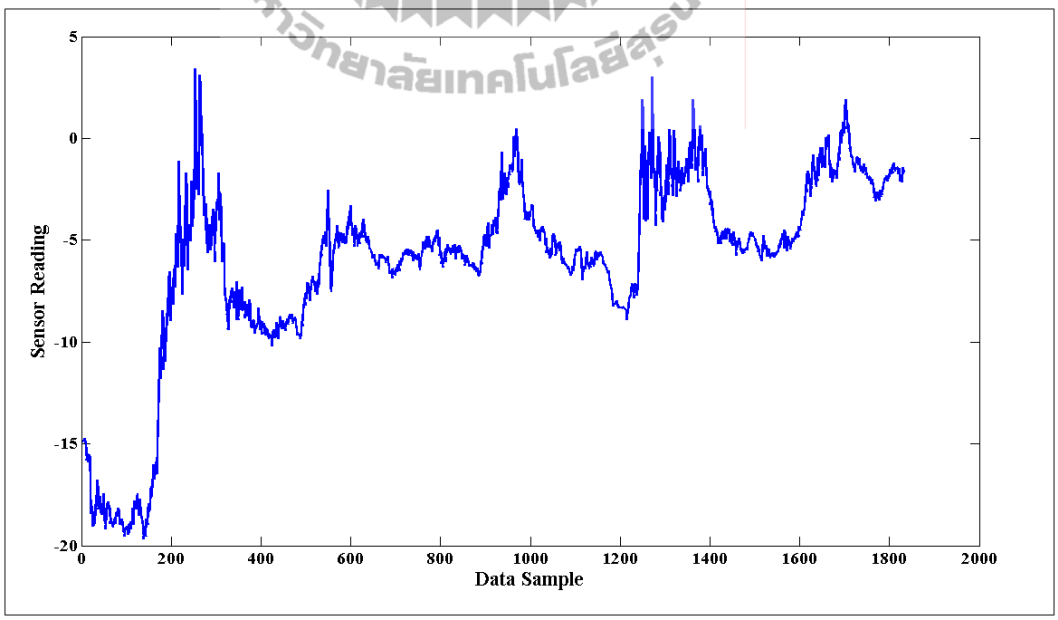


Figure B.60 LWT low-pass coefficient of pdg2008 dataset (ambient temperature reading).

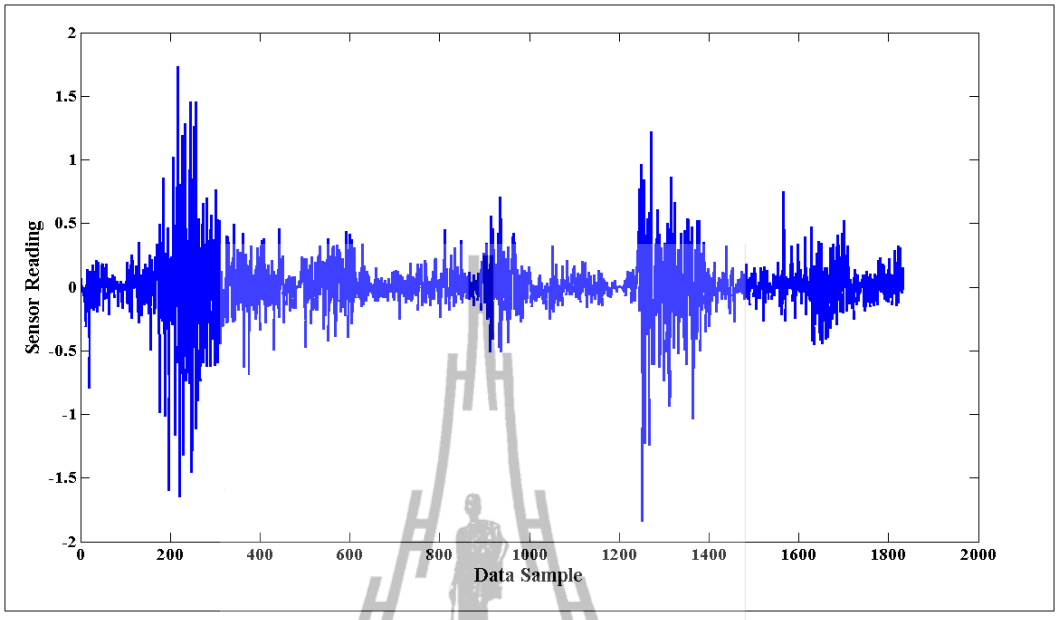


Figure B.61 LWT high-pass coefficient of pdg2008 dataset (ambient temperature reading).

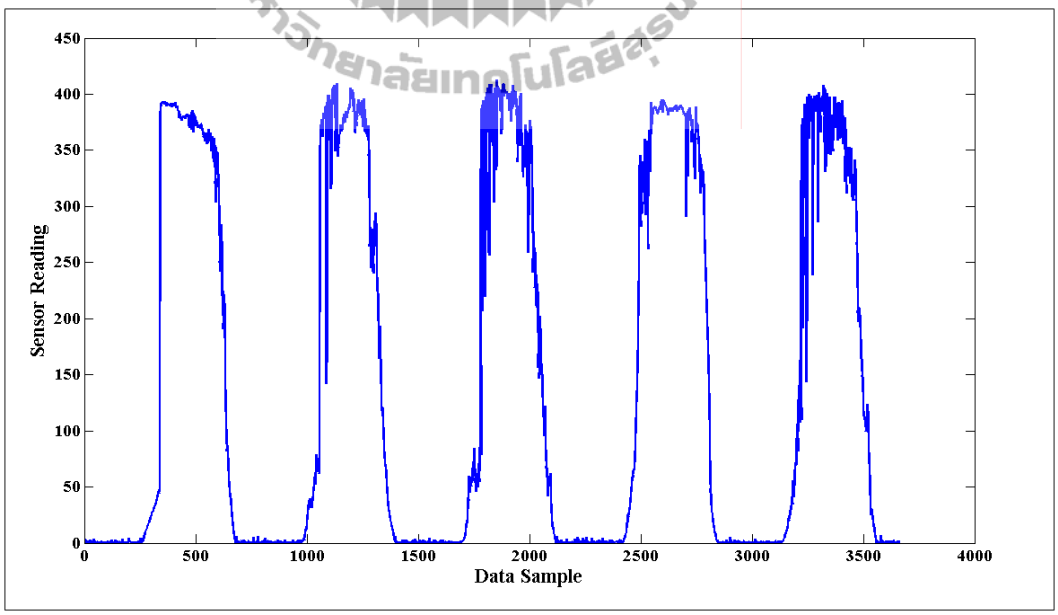


Figure B.62 pdg2008 dataset (solar radiation reading).

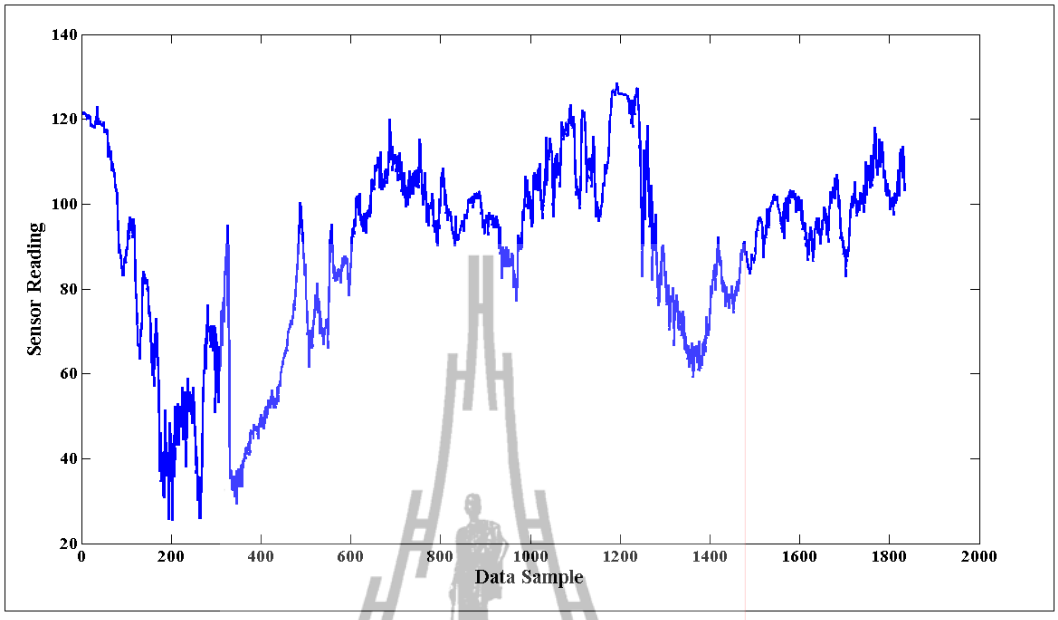


Figure B.63 DWT low-pass coefficient of pdg2008 dataset (solar radiation reading).

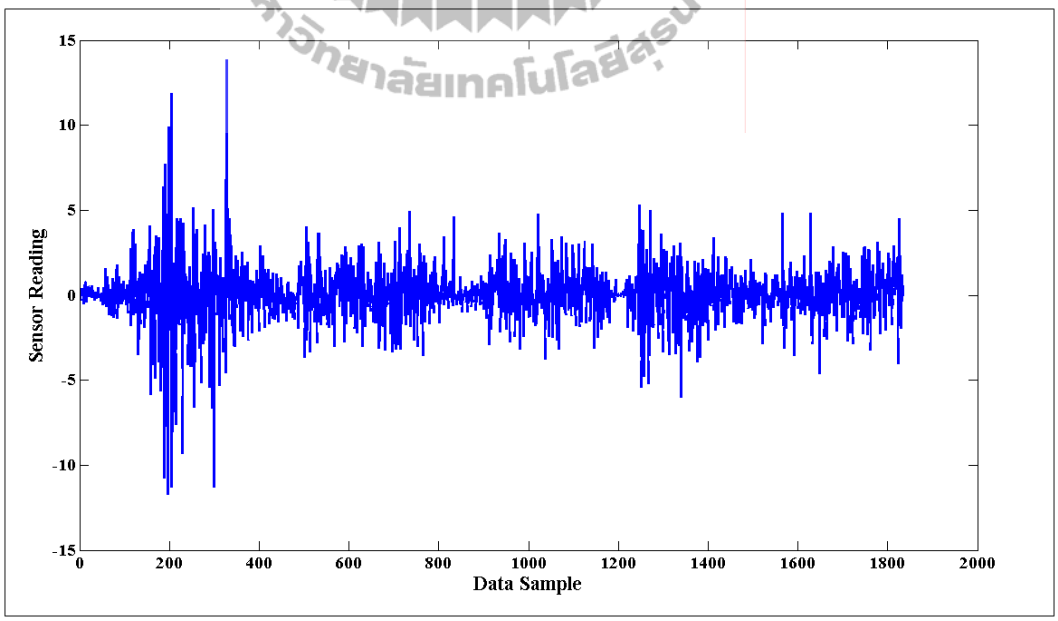


Figure B.64 DWT high-pass coefficient of pdg2008 dataset (solar radiation reading).

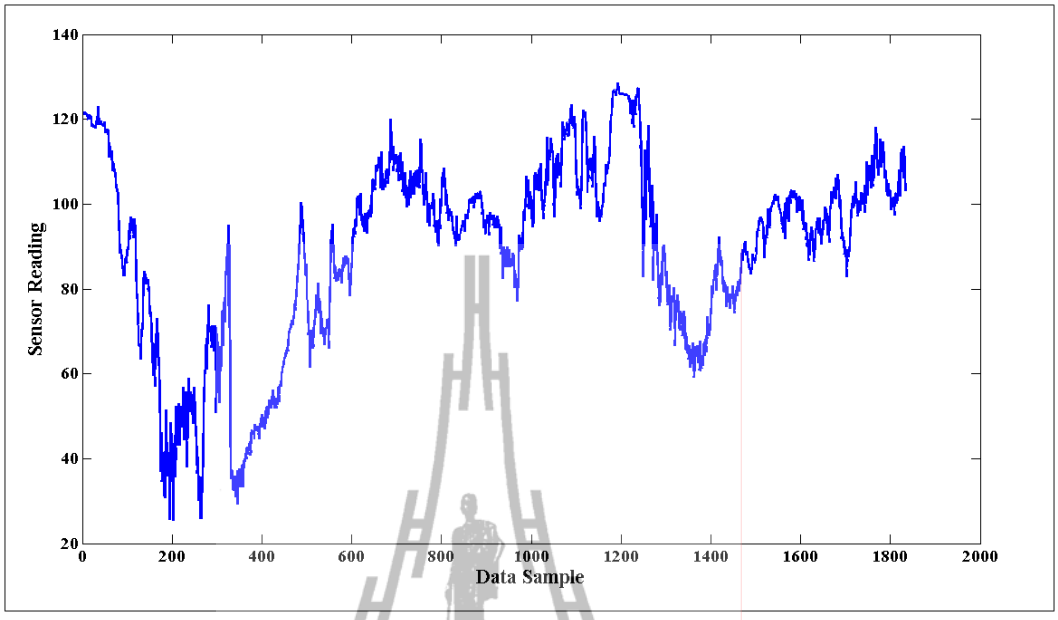


Figure B.65 LWT low-pass coefficient of pdg2008 dataset (solar radiation reading).

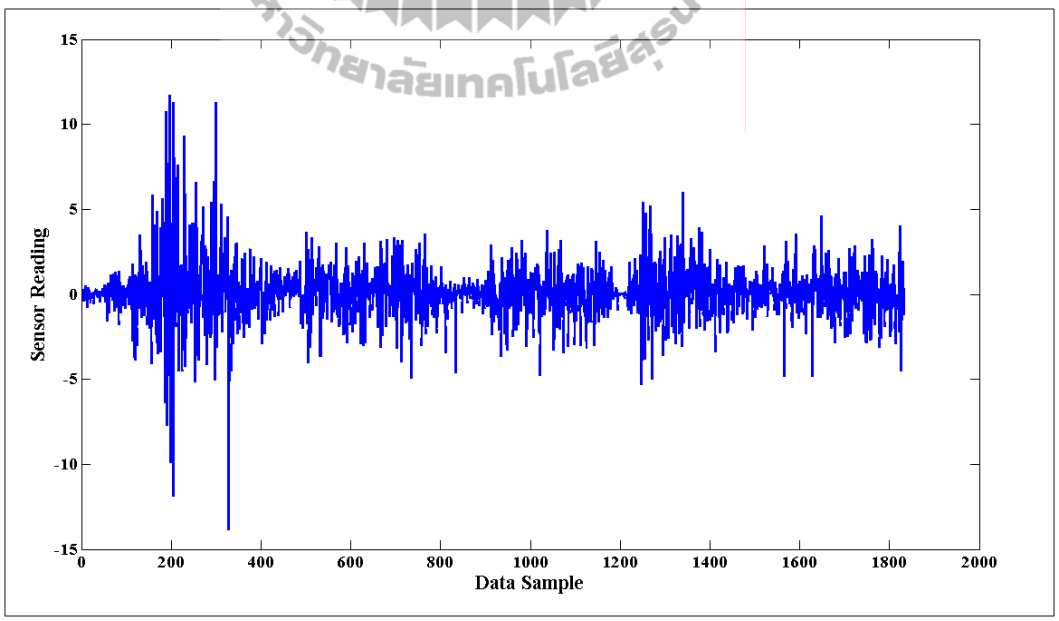


Figure B.66 LWT high-pass coefficient of pdg2008 dataset (solar radiation reading).

4. NAMOS dataset

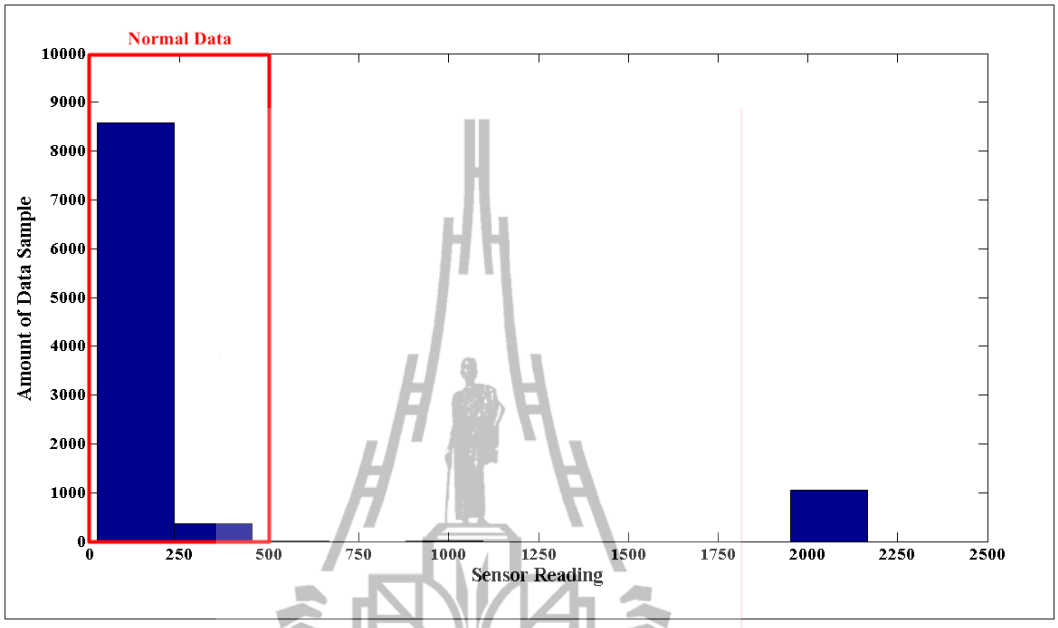


Figure B.67 Histogram of NAMOS dataset (fluorimeters reading).

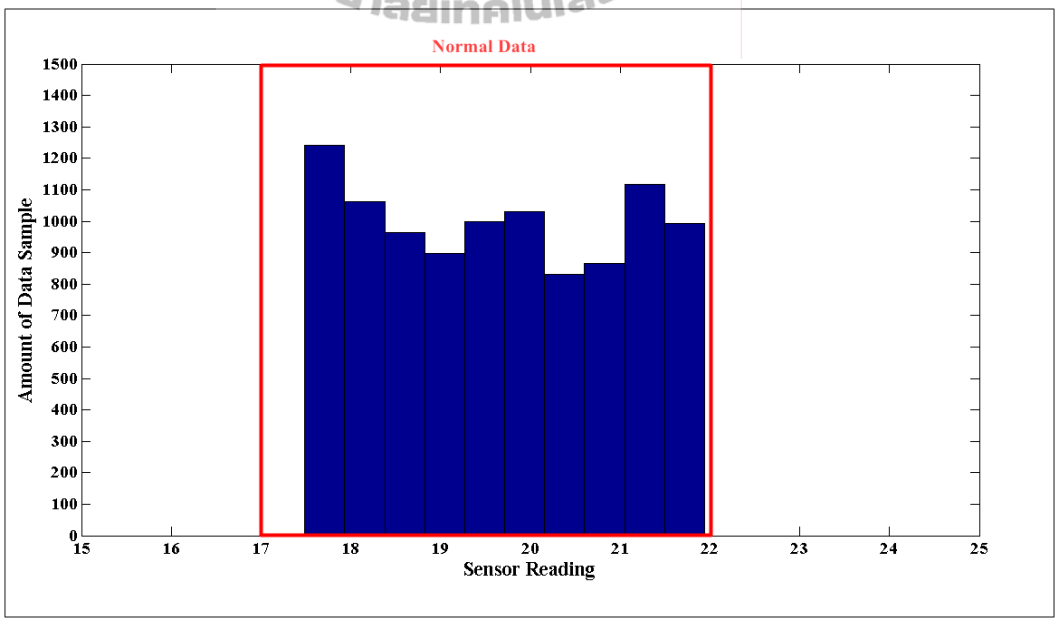


Figure B.68 Histogram of NAMOS dataset (temperature reading from sensor 1).

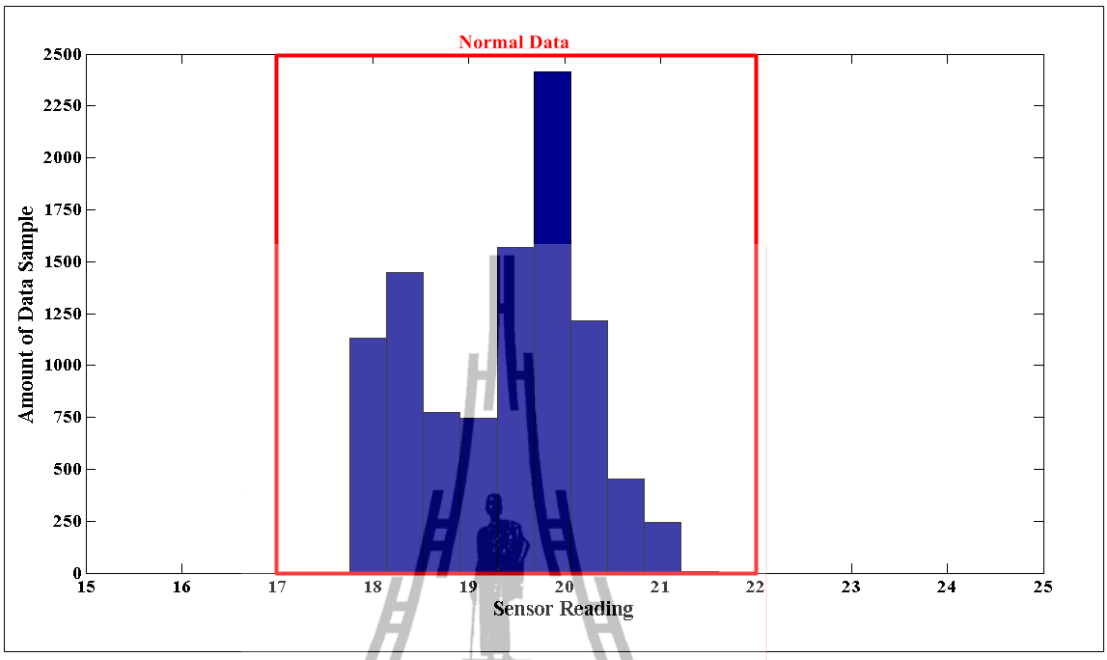


Figure B.69 Histogram of NAMOS dataset (temperature reading from sensor 2).

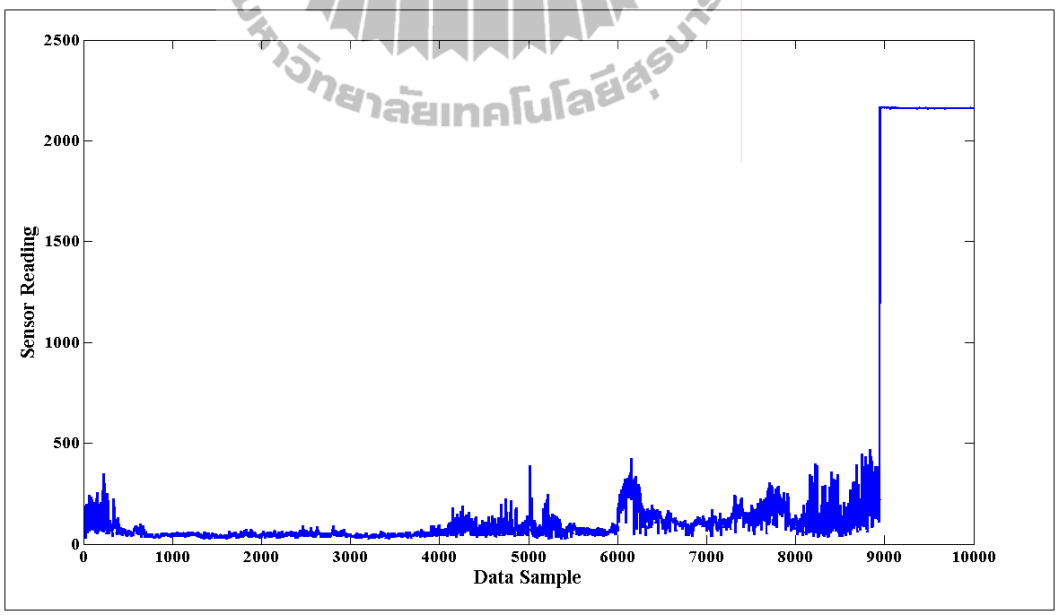


Figure B.70 NAMOS dataset (fluorimeters reading).

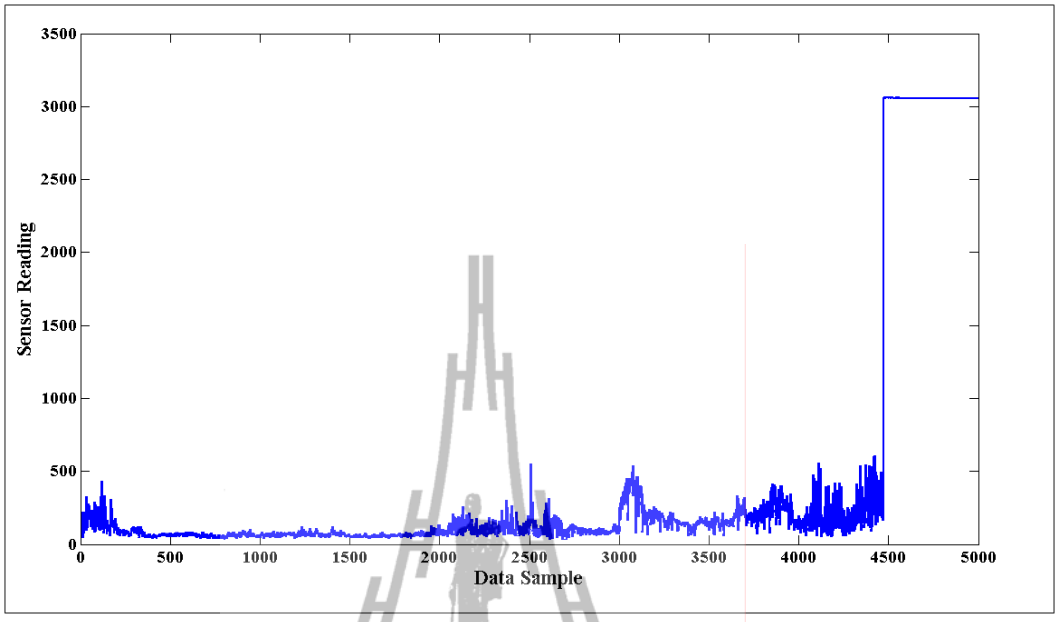


Figure B.71 DWT low-pass coefficient of NAMOS dataset (fluorimeters reading).

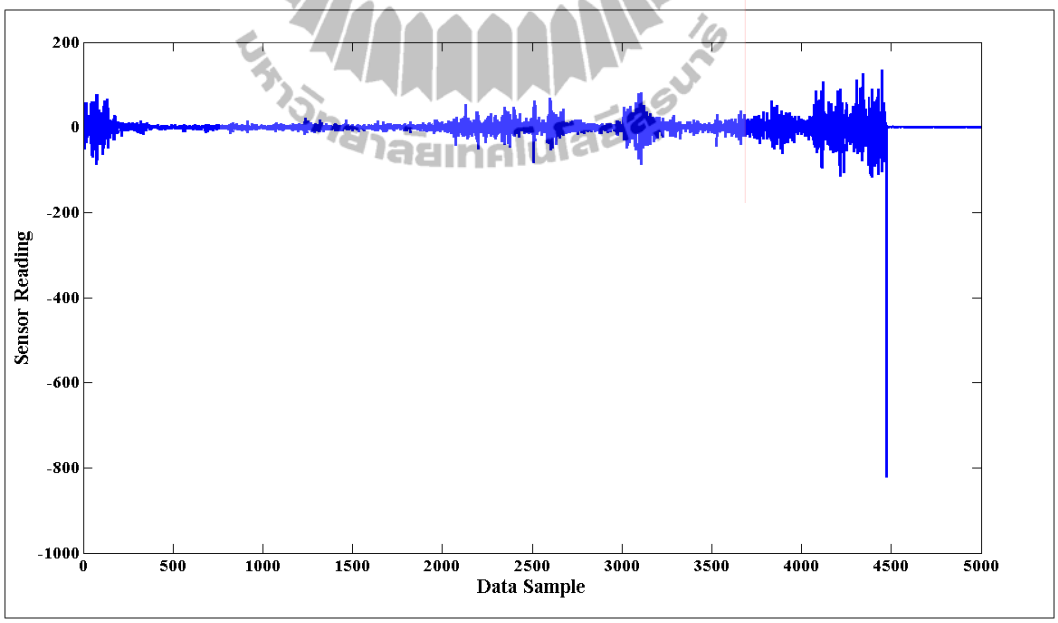


Figure B.72 DWT high-pass coefficient of NAMOS dataset (fluorimeters reading).

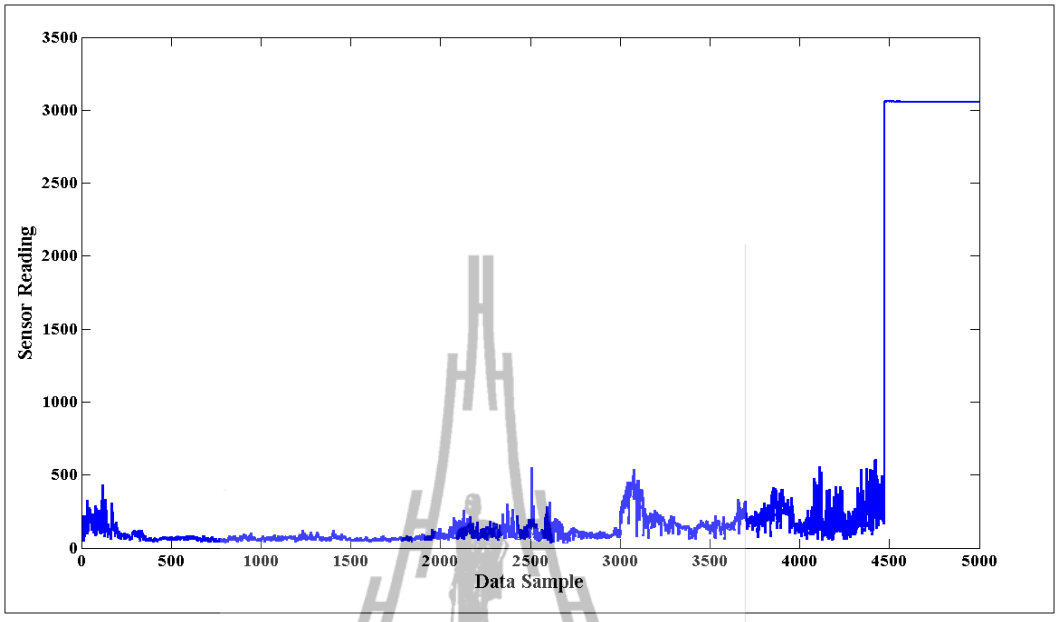


Figure B.73 LWT low-pass coefficient of NAMOS dataset (fluorimeters reading).

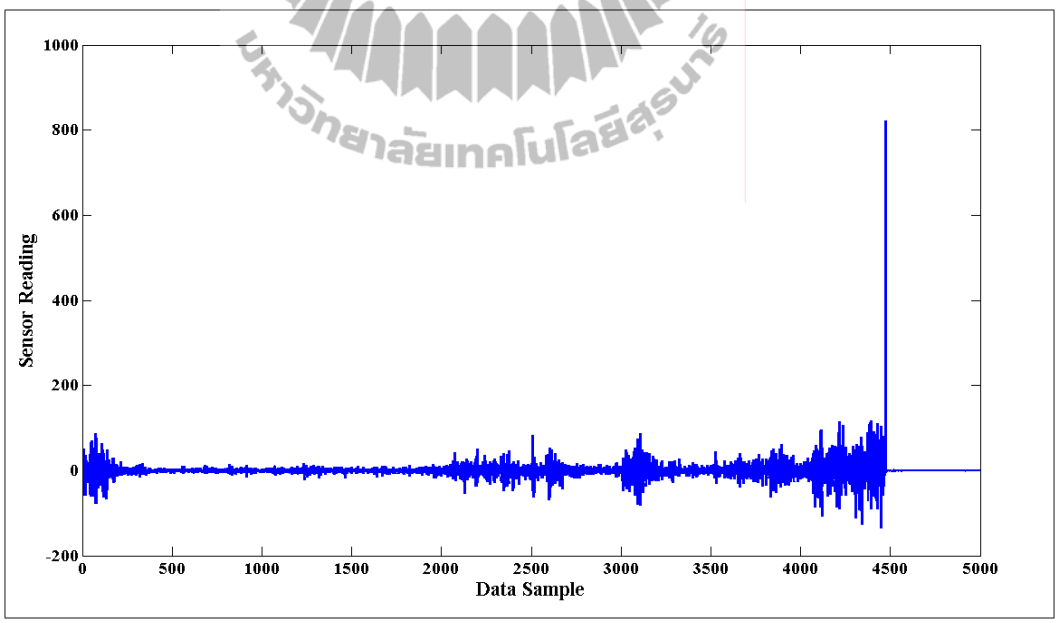


Figure B.74 LWT high-pass coefficient of NAMOS dataset (fluorimeters reading).

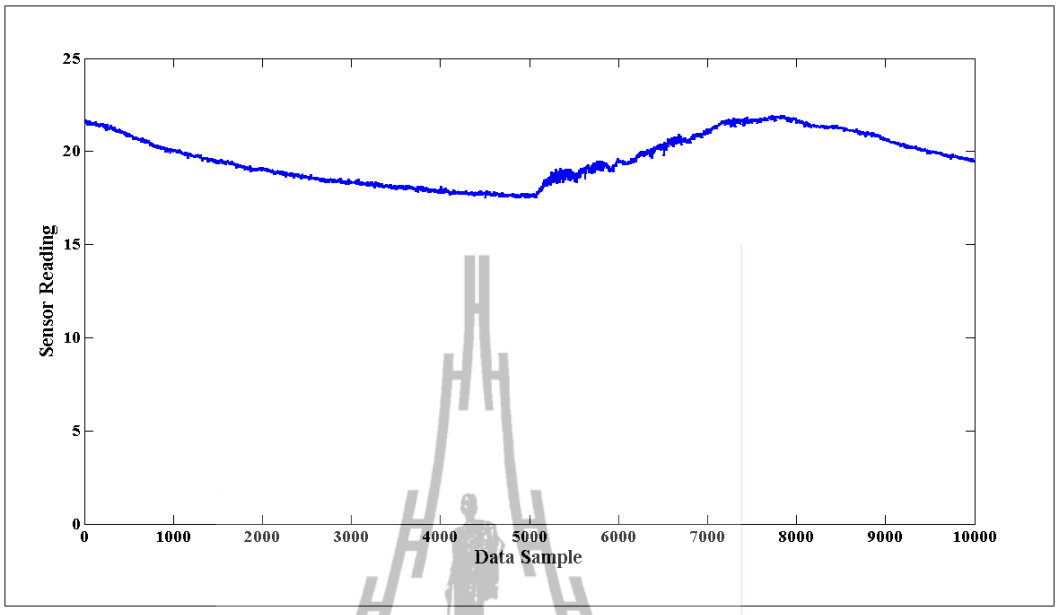


Figure B.75 NAMOS dataset (temperature reading from sensor 1).

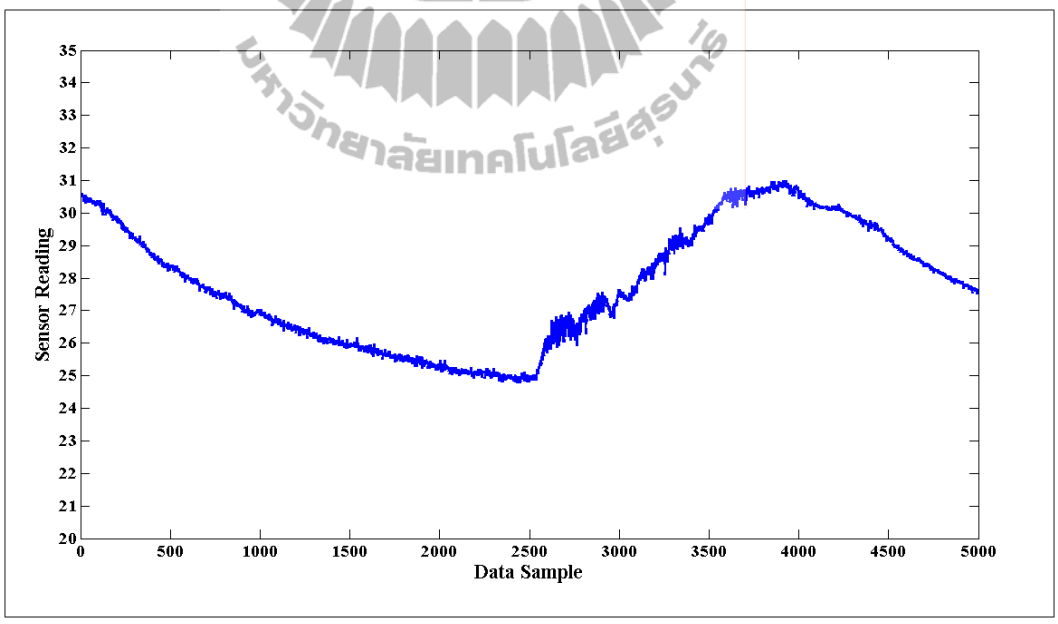


Figure B.76 DWT low-pass coefficient of NAMOS dataset

(temperature reading from sensor 1).

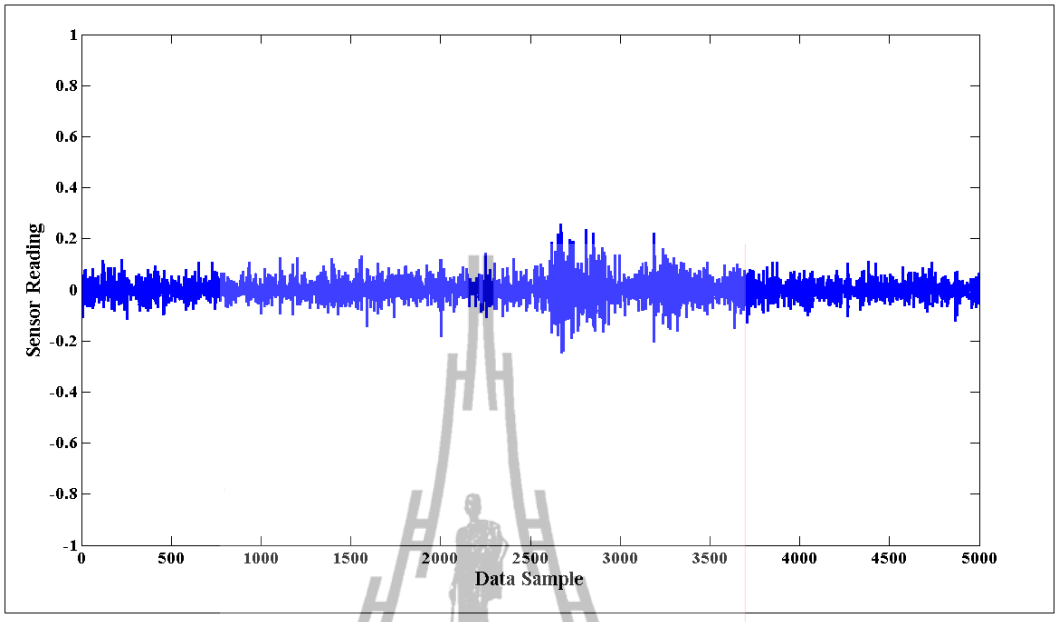


Figure B.77 DWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 1).

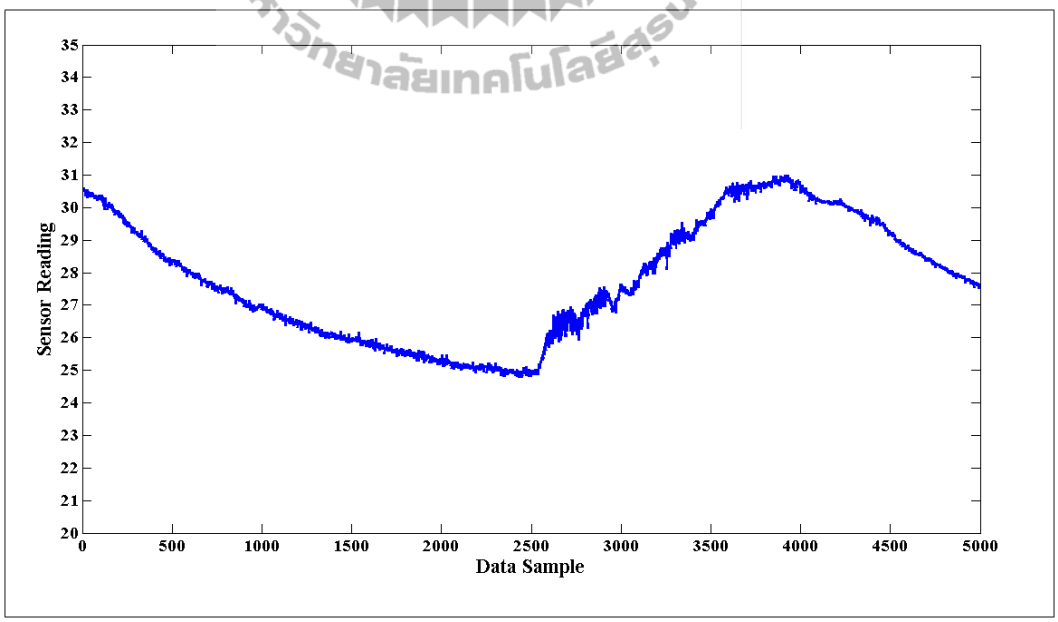


Figure B.78 LWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 1).

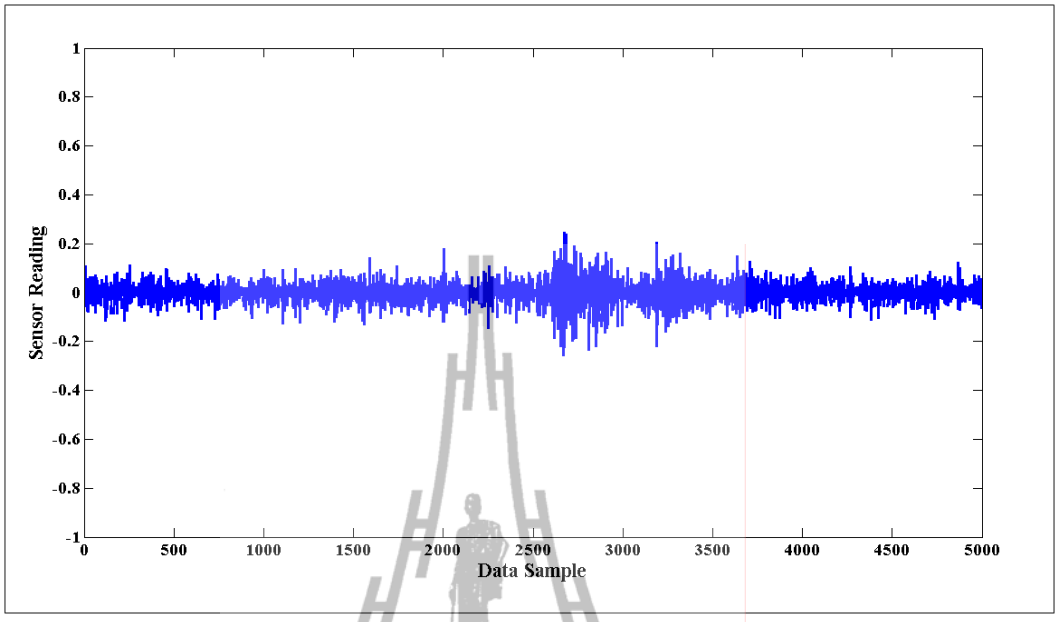


Figure B.79 LWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 1).

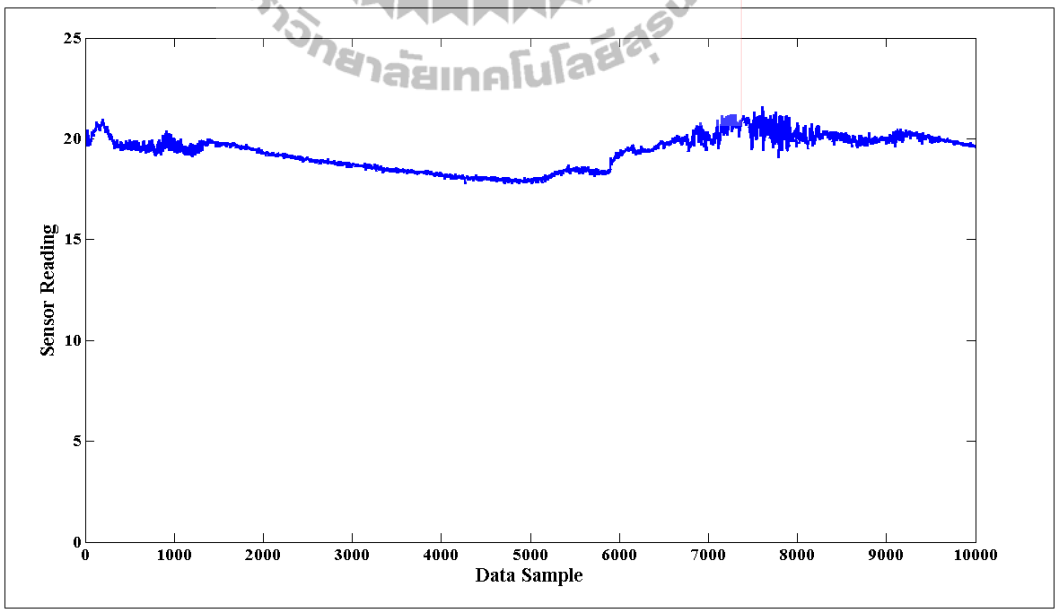


Figure B.80 NAMOS dataset (temperature reading from sensor 2).

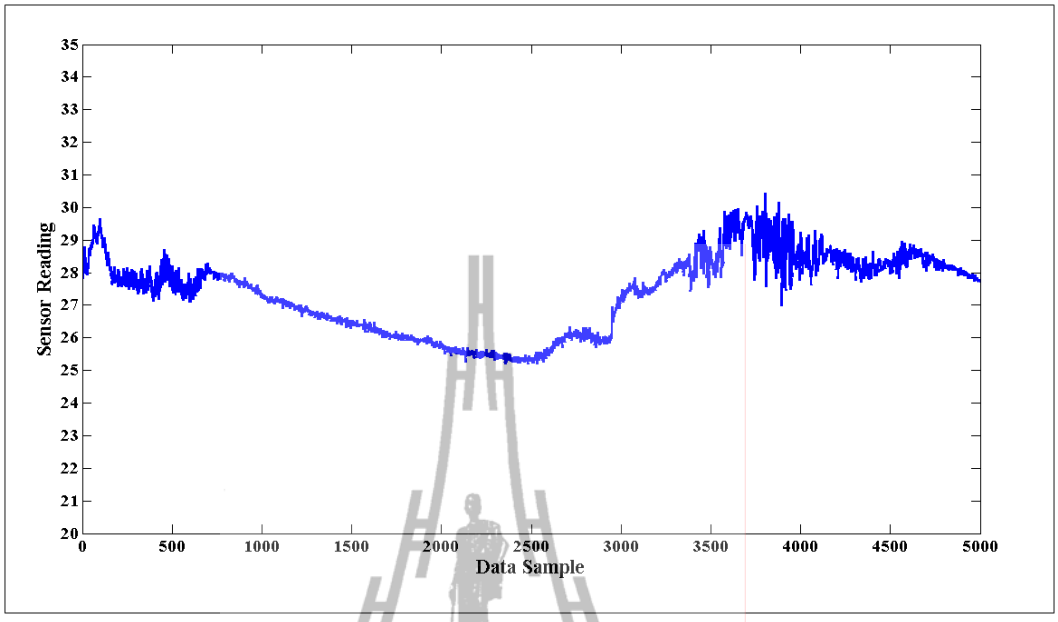


Figure B.81 DWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 2).

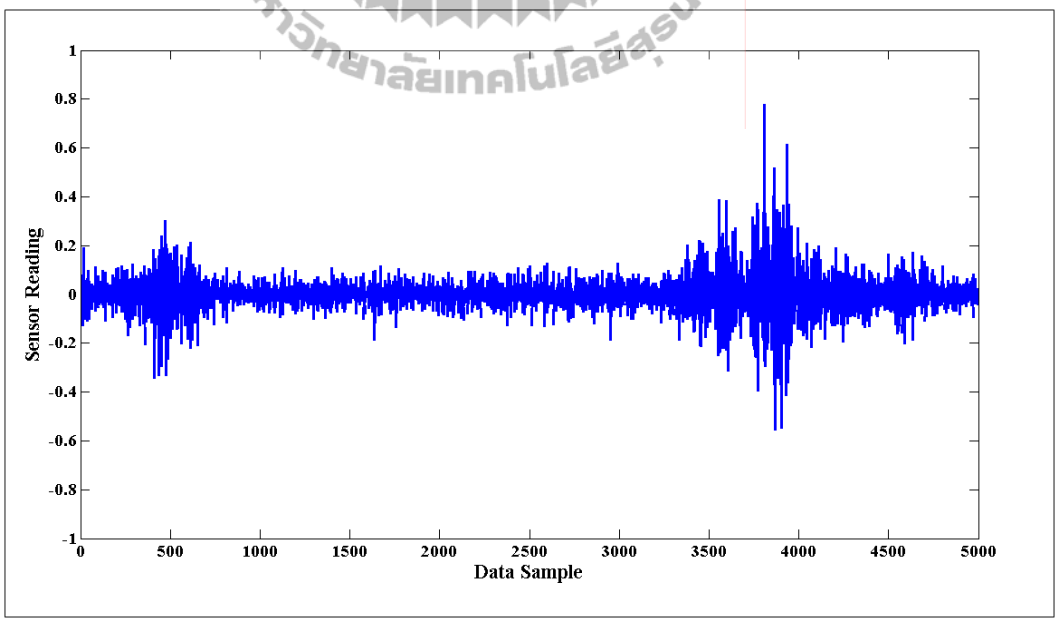


Figure B.82 DWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 2).

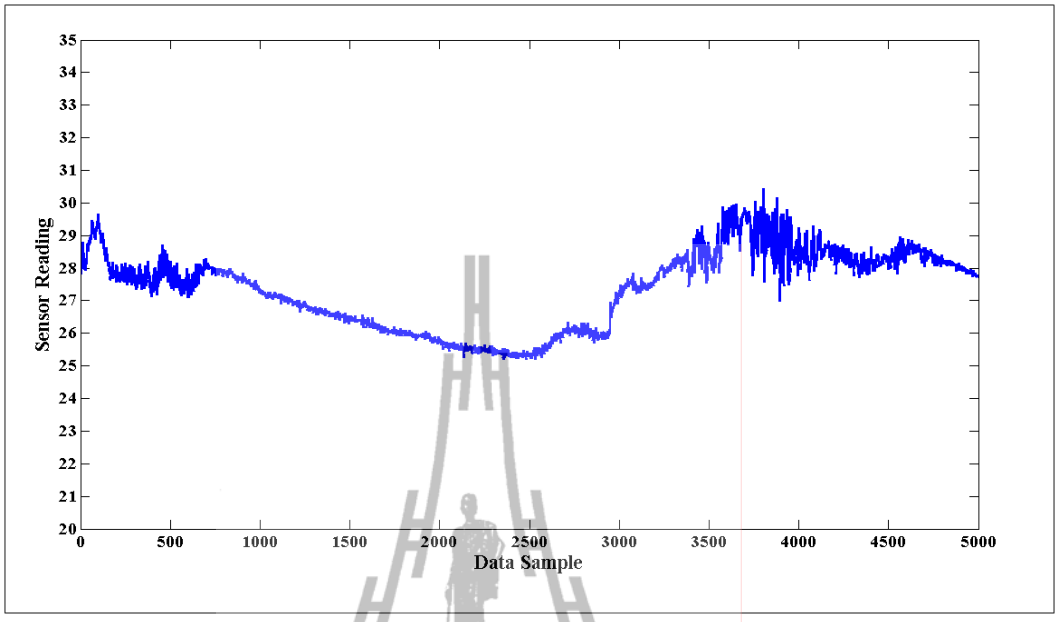


Figure B.83 LWT low-pass coefficient of NAMOS dataset (temperature reading from sensor 2).

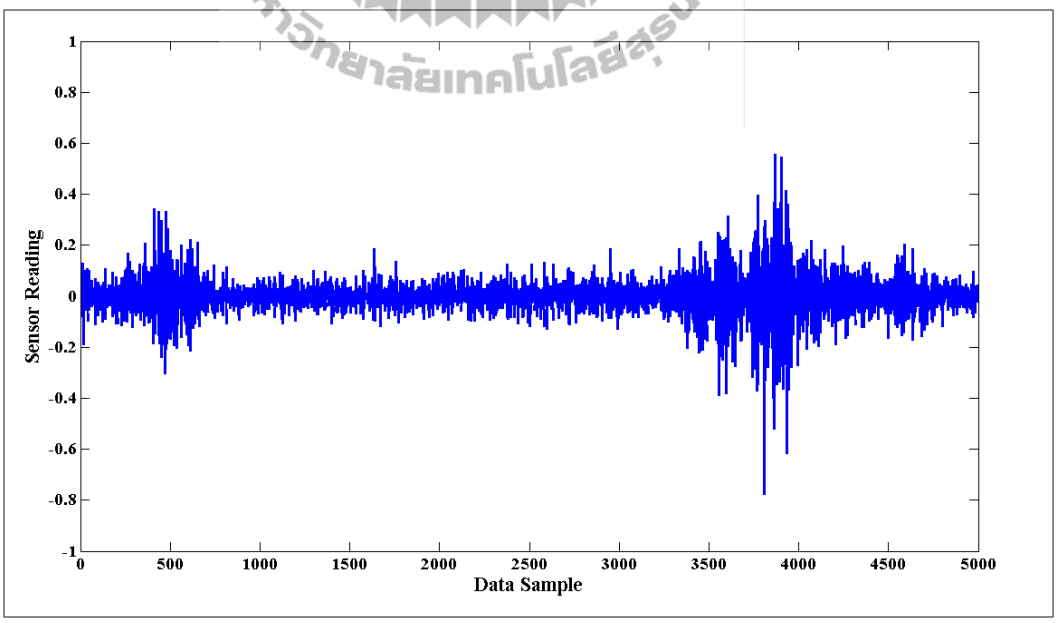
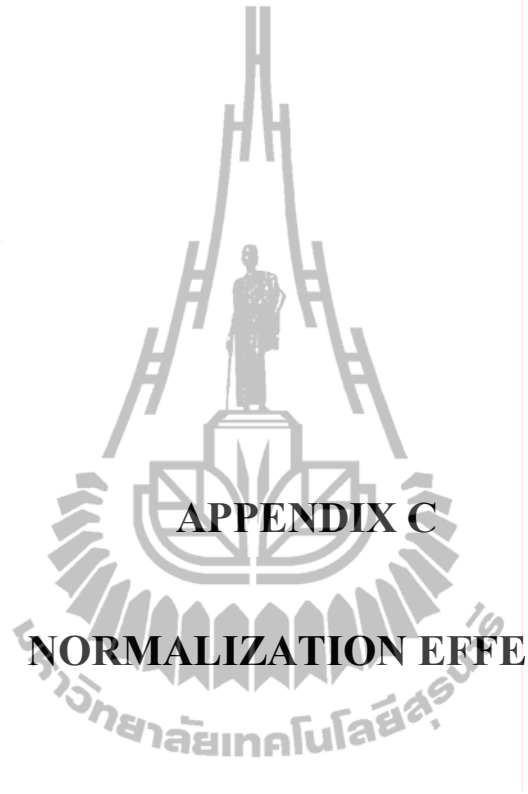


Figure B.84 LWT high-pass coefficient of NAMOS dataset (temperature reading from sensor 2).



APPENDIX C

NORMALIZATION EFFECT

Normalization Effect

In (Siripanadorn, et al., 2010a; 2010b), their synthetic data were normalized by using equation (C.1). Their synthetic data were thus normalized to a normal distribution with mean 0 and standard deviation 1

$$\frac{x_k - mean(x_k)}{\sqrt{\text{var}(x_k)}}. \tag{C.1}$$

where k is an index of KPI, x_k is a column vector of data in k th KPI.

The synthetic data in this thesis were generated from a mixture of Gaussian distributions with means randomly selected from (0.3, 0.35, 0.45) and with a standard deviation of 0.03 and normalized to the range [0, 1] using equation (C.2)

$$\frac{x_k - \min(x_k)}{\max(x_k) - \min(x_k)}. \tag{C.2}$$

In order to study the effect of normalization on anomaly detection, the synthetic data were generated from a mixture of Gaussian distributions with means randomly selected from (0.3, 0.35, 0.45) and with a standard deviation of 0.03 and normalized by both equations prior feeding to the anomaly detection algorithms.

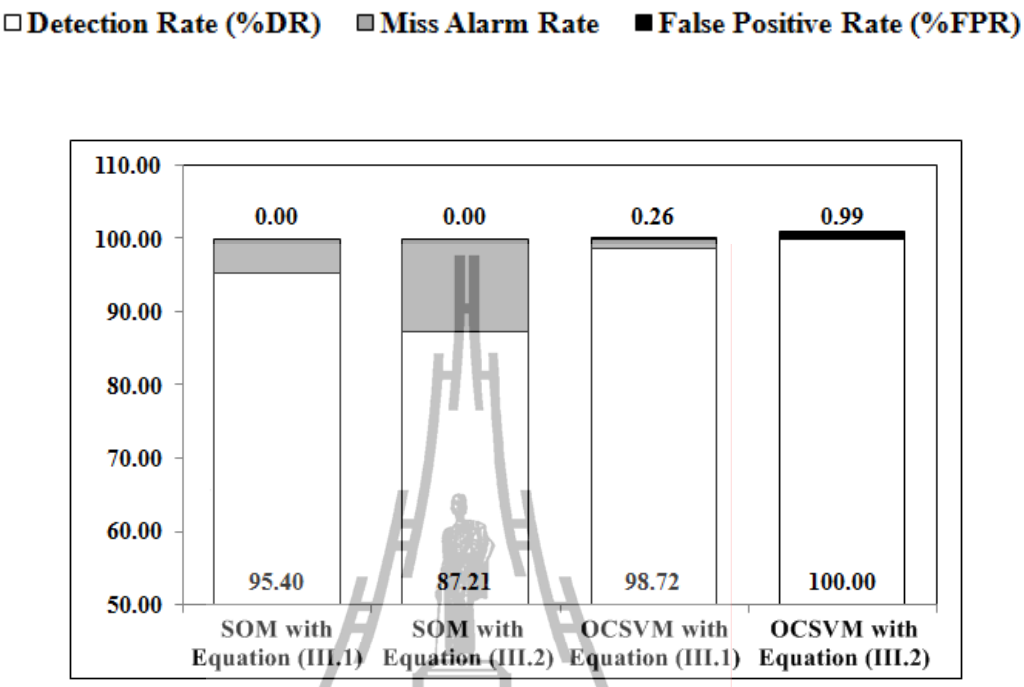


Figure C.1 Normalization effect on synthetic data with 1/80 faults.

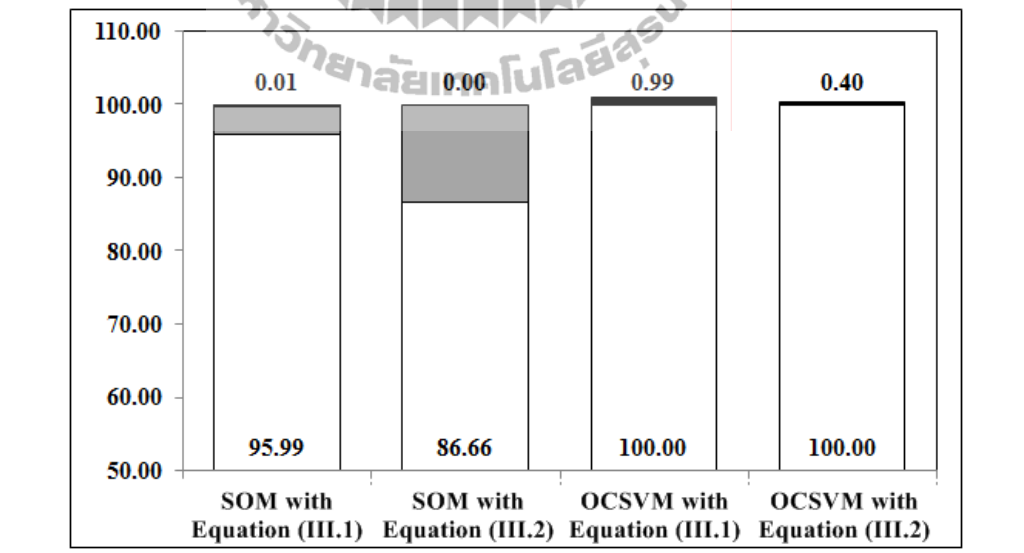
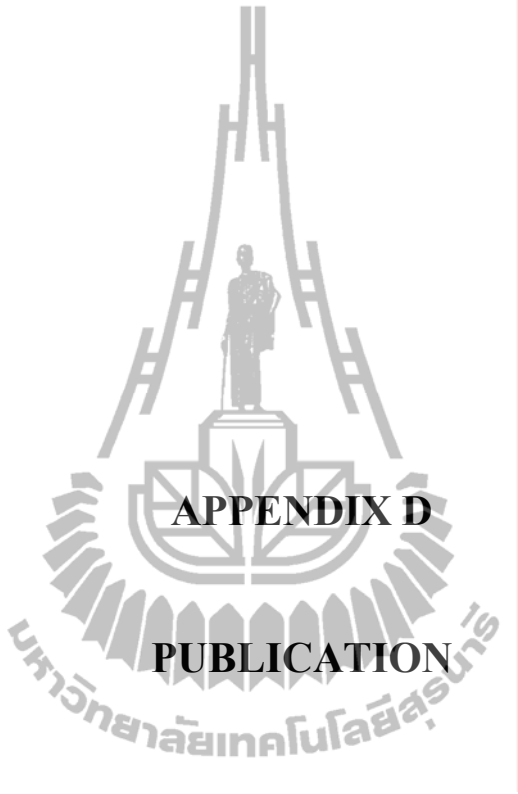


Figure C.2 Normalization effect on synthetic data with 20/4 faults.

Figure C.1, C.2 show the normalization effect on anomaly detection with synthetic data injected with 1/80 and 20/4 faults, respectively. For the SOM algorithm , normalization to normal distribution N(0, 1) using equation (C.1) gave better DR than normalization to the range [0, 1] using equation (C.2). On the other hand, OCSVM performed equally well for both types of normalizations and was therefore insensitive to the distribution of the input data. In addition, SOM with synthetic data normalization to N(0, 1) perform equally well as OCSVM. Therefore, SOM algorithm is suitable for dataset with normal distribution. However, datasets may not always be normally distributed, and consequently, affect SOM's performance. Therefore, in this thesis, a normalization method irrespective of data distribution such as in equation (C.2) was selected to preprocess the data prior to anomaly detection.



APPENDIX D

PUBLICATION

Publication

Takianngam, S., and Hattagam, W., (2011). Discrete Wavelet Transform and One-Class Support Vector Machines for anomaly detection in wireless sensor networks, **Proceedings of International Symposium on Intelligent Signal Processing and Communications Systems.** (Published)

2011 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) December 7-9, 2011

Discrete Wavelet Transform and One-Class Support Vector Machines for Anomaly Detection in Wireless Sensor Networks

Saowaluk Takianngam
School of Telecommunication Engineering,
Institute of Engineering,
Suranaree University of Technology,
Nakhon Ratchasima, Thailand
stk.pupae@gmail.com

Wipawee Usaha
School of Telecommunication Engineering,
Institute of Engineering,
Suranaree University of Technology,
Nakhon Ratchasima, Thailand
wusaha@ieee.org

***Abstract*— Data readings from wireless sensor networks (WSNs) may be abnormal due to detection of unusual phenomena, limited battery power, 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. This paper proposes an integrated data compression and anomaly detection algorithm in WSNs which can detect anomalies accurately by employing half of sensor data measurement, instead of using all the sensor data measurement. The contribution of this paper centers on data compression by using Discrete Wavelet Transform (DWT) then feeding to anomaly detection by using One-Class Support Vector Machine (OCSVM). We tested our algorithm with several synthetic and real world datasets. The results showed that the proposed algorithm outperformed previous techniques in terms of near 100% detection rate and with marginal increase in false positive rates in presence of short and noise faults.**

Keywords-discrete wavelet transform; self-organizing map; one-class support vector machine; data compression; anomaly detection; wireless sensor networks

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of wireless sensor nodes located at different places in an area of interest. Data measurements are collected by these sensor nodes and forwarded to a central server. WSNs are formed using many sensor nodes that have many limitations such as memory, bandwidth, energy consumption, and computational capabilities [1]. These limitations make communication unreliable which can contribute to occurrences of the anomalies in a set of sensor data measurements.

An anomaly or outlier in a set of sensor data measurements is defined as an observation that appears to be inconsistent with the remainder of the dataset [2]. Anomalies, which occur from unusual phenomena in monitor domain, can damage agricultural produce. Some applications, such as in a hydroponics farm that requires accurate pH level control of solution plant, or in a bio-organic fertilizer plant that requires temperature control in the fertilizer compost process, immediate anomaly detection in a set of data measurement is essential in order to take immediate course of actions.

However, due to hardware limitations WSNs require minimal energy consumption. Since radio communication in WSN consumes more energy than processing and computing [2], some researches such [3], [4] used data compression by Discrete Wavelet Transform (DWT) prior to feeding data to an anomaly detection algorithm. Such approach was found to increase the efficiency of anomaly detection. Motivated by their findings, we extend their study to integrate DWT data compression with an alternative anomaly detection technique.

The One-Class Support Vector Machine (OCSVM) [5] is a popular and useful anomaly detection technique that does not assume any prior knowledge about the distribution of the data and has been found suitable for resource-constrained WSNs [5].OCSVM can update the normal behavioral model of the sensed data in an online manner. References [6], [7], [8] and [9] successfully used OCSVM to detect anomalies in WSNs, with real world datasets based on fitting normal data to a quarter of sphere feature space that can change in dynamic environment. However, to the best of our knowledge, the integration between OCSVM and DWT has not yet been proposed. Therefore, the underlying aim of this paper is to study the effect and efficiency of DWT data compression on the OCSVM anomaly detection technique and assess its suitability for deployment in resource-constrained conditions in WSNs.

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 $x^{\mu} \in \Re,\ \mu = 1, \ldots, n$

$$x^{\mu} = [x_1^{\mu}, x_2^{\mu}, \ldots, x_n^{\mu}] = [KPI_1^{\mu}, KPI_2^{\mu}, \ldots,\ KPI_n^{\mu}] \quad (1)$$

978-1-4577-2166-3/11/$26.00 ©2011 IEEE

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. One-Class Support Vector Machine (OCSVM)

Tax and Duin [10] have proposed a one-class support vector machine (OCSVM) formulation for outlier detection. Then Laskov [7] have extended this approach into a special type of SVM call *Quarter-Sphere OCSVM*. The key idea of this algorithm is to encompass the data with a hypersphere anchored at the center of mass of the data in feature space. Here we provide the mathematical formulation of the one-class quarter-sphere SVM.

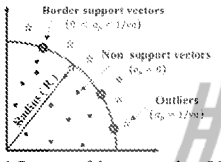


Figure 1 Geometry of the quarter-sphere OCSVM

Consider an input dataset $X = \{x^\mu : \mu = 1, \dots, n\}$ of p variate data vector $x^\mu = (x_1^\mu, x_2^\mu, \dots, x_p^\mu), \mu = 1, \dots, n$ in the input space $\Re^p$ where the number of data vector in a dataset X is n. In principle, X is mapped to a feature space $\Re^q$ via a nonlinear function $\phi(\cdot)$, resulting in a set of the image vectors $X_\phi = \{\phi(x^\mu) : \mu = 1, \dots, n\}$ where a row vector of image vectors is $\phi(x^\mu) = \left(\phi(x_1^\mu), \phi(x_2^\mu), \dots, \phi(x_q^\mu)\right), \mu = 1, \dots, n$. The aim is to fit a hypersphere in a feature space with minimum effective radius $R(> 0)$, centered at the origin, encompassing a majority of the image vectors X_ϕ. This can be formulated as an optimization problem as follows:

$$\min_{R \in \Re^+, \xi \in \Re^n} R^2 + \frac{1}{\nu n} \sum_{\mu=1}^{n} \xi_\mu$$

$$\text{Subject to :} \quad k(x^\mu, x^\mu) \le R^2 + \xi_\mu$$

$$\xi_\mu \ge 0 \qquad (2)$$

where $\{\xi_\mu : \mu = 1, \dots, n\}$ are the slack variables that allow some of the image vectors to lie outside the sphere. The parameter $\nu \in (0, 1)$ is the regularization parameter which controls the fraction of image vectors that lie outside the sphere, i.e., the fraction of image vectors that can be anomalies. Note that $k(x^\mu, x^\mu) = \phi(x^\mu) \cdot \phi(x^\mu)^{\mathrm{T}}$ for a Mercer kernel and $k(x^\mu, x^\mu)$ is a kernel function which was used to compute the similarity of any two vectors in the feature space using the original attribute set. The dual formulation of this primal problem (2) can be obtained as follows:

$$\min_{\alpha \in \Re^n} -\sum_{\mu=1}^{n} \alpha_\mu k(x^\mu, x^\mu)$$

$$\text{Subject to :} \quad \sum_{\mu=1}^{n} \alpha_\mu = 1$$

$$0 \le \alpha_\mu \le \frac{1}{\nu n} \qquad (2)$$

where $\alpha_\mu \ge 0$ is a Lagrangian multiplier, $\mu = 1, \dots, n$. This dual problem (2) is a linear optimization problem. In order to alleviate this problem, the image vectors in the feature space are centered in the space using center kernel matrix as follows:

$$K_c = K - 1_n K - K 1_n + 1_n K 1_n \qquad (3)$$

where K is a $n \times n$ kernel metric consist of $k(x^\mu, x^\omega)$ where $\mu, \omega = 1, \dots, n$. If $\mu = \omega$, we can obtain $k(x^\mu, x^\omega) = k(x^\mu, x^\mu) = k(x^\omega, x^\omega)$ and can obtain from the norms of image vector $\phi(x^\mu)$. Otherwise, $k(x^\mu, x^\omega)$ and can obtain from the kernel function. Furthermore, 1_n is a $n \times n$ metrix with all values equal to $1/n$. Once the image vectors are centered, the norms of the kernels are no longer equal. Hence the dual problem (2) can now be solved.

The $\{\alpha_\mu\}$ can be obtained using widely available linear optimization techniques. The image vectors can be classified as Figure 1. The image vectors with $\alpha_\mu = 0$ will fall inside the sphere. The image vectors with $\alpha_\mu > 0$ are called the *support vectors*. Support vectors with $\alpha_\mu = 1/\nu n$ are termed as *outliers*, which fall outside the sphere. Support vectors with $0 < \alpha_\mu < 1/\nu n$ will reside *on* the surface of the sphere, and hence are called the *border support vectors*. Moreover, the radius of the sphere R can be obtained using $R^2 = k(x^\mu, x^\mu)$, for any border support vector x^μ.

B. Self-Organizing Map (SOM)

Competitive neural models such as the self-organizing map (SOM) 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.

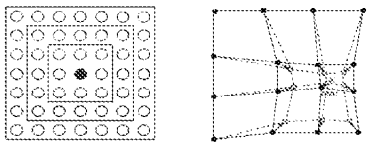


Figure 2 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3 neighborhoods, (b) SOM updates the BMU with 1- neighborhood.

The basic SOM consists of a regular grid of map units or neurons as shown in Figure 2 (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 (a).

Neuron i maintains a weight vector m_i. In order to follow the properties of the input data, such vector is updated during the training process. For example, Figure 2 (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 $x = \{x^{\mu}: \mu = 1, \dots, s\}$ from the input dataset $X = \{x^{\mu}: \mu = 1, \dots, n\}$ is chosen where the number of sample vector x is s and the number of data vector in a dataset X is n. 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 $m_i(t)$ is the closest to the input vector $x^{\mu}(t)$. We denote such weight vector by $m_c(t)$ and refer to it as the Best-Matching Unit (BMU), which is

$$\|x^{\mu}(t) - m_c(t)\| = \arg\min_{\forall i} \|x^{\mu}(t) - m_i(t)\| \quad (4)$$

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

$$m_i(t+1) = m_i(t) + \eta_t h_c(i,t)[x^{\mu}(t) - m_i(t)] \quad (5)$$

where t is the iteration index, $x^{\mu}(t)$ 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

$$h_c(i,t) = exp\left[-\frac{\|r_c(t) - r_i(t)\|^2}{2\sigma^2(t)}\right] \quad (6)$$

here $r_i(t)$ 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 $\lim_{t\to\infty} \eta_t = 0$ and $\lim_{t\to\infty} h_c(i,t) = 0$ for the algorithm to converge [3].

III. Data Compression

A. Discrete Wavelet Transform (DWT)

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:

$$a_{j+1}^{DWT}(f) = \sum_n h_0(n-2f) a_j^{DWT}(f) \quad (7)$$

$$d_{j+1}^{DWT}(f) = \sum_n g_0(n-2f) a_j^{DWT}(f) \quad (8)$$

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, f 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} [3].

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.

IV. Experiment Results

This section consists of two parts. First, we evaluated the performance of the proposed integration of DWT and OCSVM algorithm by detecting anomalies in series of synthetic data and real world datasets. We then proceed to evaluate the performance of a previous technique using DWT and SOM [3], [4] in comparison to our algorithm.

A. Datasets for Experiment

We categorized faults into 3 types as shown in Figure 3, i.e., noisy faults, short faults and constant faults [11]. A noisy fault is a fault that occurs when variance of the sensor readings increases and affects a number of successive samples. A short fault is a sharp change in the measurement value between two successive data points and affects a single sample at a time. A constant fault is a fault that occurs when a constant value for a large number of successive samples is reported.

We used both synthetic and real world datasets. Three real-world datasets were used for the performance evaluation, namely, INTEL [12], SensorScope [13], and NAMOS [14] datasets.

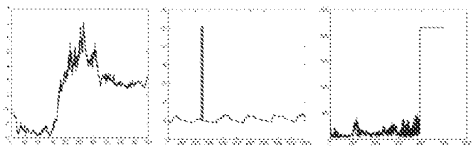


Figure 3 Fault in sensor reading
(a) noisy faults, (b) short faults and (c) constant faults.

1) Synthetic Data: The synthetic data is generated from a mixture of Gaussian distributions with means randomly selected from (0.3, 0.35, 0.45) and with a standard deviation of 0.03 using MATLAB. Data was generated for 15 sensor nodes and two features of 106 data vectors per sensor node. The combined data comprised 1590 data vectors. Then we introduced a number of faults uniformly distributed ranging between [0.50,1] to each feature of the data. 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 noisy fault which was 20/4, then 10/8, 5/16, 2/40, and finally to short faults which was 1/80. All these types of faults gave a total of 80 faulty data. The whole dataset was normalized to the range [0, 1]. The exact positions of the faults injected in the input data were predetermined and was later used to detect true and false positive alarms.

2) INTEL: 54 Mica2Dot motes with temperature, humidity and light sensors were deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004 [12]. 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. By considering the positions of anomalous data, we found that this dataset had short faults.

3) SensorScope: This dataset is available in [13]. It consists of two sub datasets as follow:

a) station no.39 (SS39): In this experiment, we presented the results on anomaly detection in one KPI of SensorScope which was collected from weather station no.39 (SS39). Using visual inspection and the histogram method, the lower and upper threshold valued used for anomaly detection were 1.5 and 9. By considering the positions of anomalous data, we found that this dataset depicted short faults.

b) pdg2008-metro-1: In the experiment, we used two types (KPIs) of data in the pdg2008-metro-1 dataset for anomaly detection, 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 were -14 and 4 for the surface temperature and -12 and 4 for the ambient temperature. By considering the positions of anomalous data, we found that this dataset contained noisy faults.

4) NAMOS: In this dataset, 9 buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, for over 24 hours in August, 2006 [14]. We analyzed the measurements from chlorophyll sensors on buoys no. 103 for 10^4 samples. 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. By considering the positions of anomalous data, we found that constant faults were present in this dataset.

B. Evaluation of DWT with OCSVM

In this paper, we use the linear kernel as the distance based kernel. The linear kernel function for data vectors x^{μ} and x^{ω} is given by $k_{linear}(x^{\mu}, x^{\omega}) = \phi(x^{\mu}) \cdot \phi(x^{\omega})$

In each simulation, we recorded the false positives, which occurred when a normal measurement was identified as anomalous by the detector, and the true positives, which occurred when an actual anomalous measurement was correctly identified by the detector. The false positive rate (FPR) was computed as the percentage ratio between the false positives and the actual normal measurements, and the detection rate (DR) was computed as the percentage ratio between the true positives and the actual normal measurements.

In our proposed integration of DWT with OCSVM (OCSVM+DWT) algorithm, we improved the performance of the OCSVM part of the algorithm by replacing the original set of input data with low pass or high pass DWT coefficients by using Haar mother wavelet. The low (high) pass DWT coefficients obtained from DWT were referred as *low (high) pass data* with just half of the original data size, whereas the original data were referred as *uncompressed data*.

Figure 4 shows the receiver operating characteristics (ROC) curve obtained for the OCSVM+DWT schemes for various datasets by varying the ν parameters from 0.02 to 1 in increments of 0.02. The results showed that ν significantly affected DR and FPR in OCSVM. The value ν is the fraction of detected outliers [8], and therefore is directly proportional to the radius R (see Figure 1). Hence, the greater ν value, the more the outliers detected (thus the higher DR and FPR). Figures 4(a) shows the synthetic dataset injected short faults results. Note that all algorithms performed equally well. Figures 4 (b), (c) and (d) show results for the synthetic dataset injected with noise faults. Note that the OCSVM+DWT (LP) performed equally well in terms of DR as the OCSVM alone with uncompressed data. However, the OCSVM+DWT with high pass data gave the worst performance. This was because HP coefficients reflect the rate of changes between two successive samples. Therefore, HP coefficients were more suitable for short faults whereas LP coefficients were more suitable for slower changing faults like noise faults. Figures 4 (e), (f), (g) and (h) show the real world dataset results. Figure 4 (e) shows that the Intel dataset and the SS39 datasets gave 100% DR for all algorithms. This was because the Intel and the SS39 dataset contain short faults which with high amplitude which can be easily detected. Figures 4 (g) and (h) illustrate the results for the pdg2008-metro-1 and the NAMOS datasets. Note that OCSVM+DWT (LP) obtained higher DR and lower FPR, thereby outperforming the OCSVM alone with uncompressed data and OCSVM+DWT (HP). This was because the pdg2008-metro-1 dataset contained noise faults and the NAMOS dataset contained constant fault. Both types of faults were trend-like changes and therefore were more significant when captured with LP coefficients than HP coefficients. Therefore, OCSVM+DWT (HP) data performed worst.

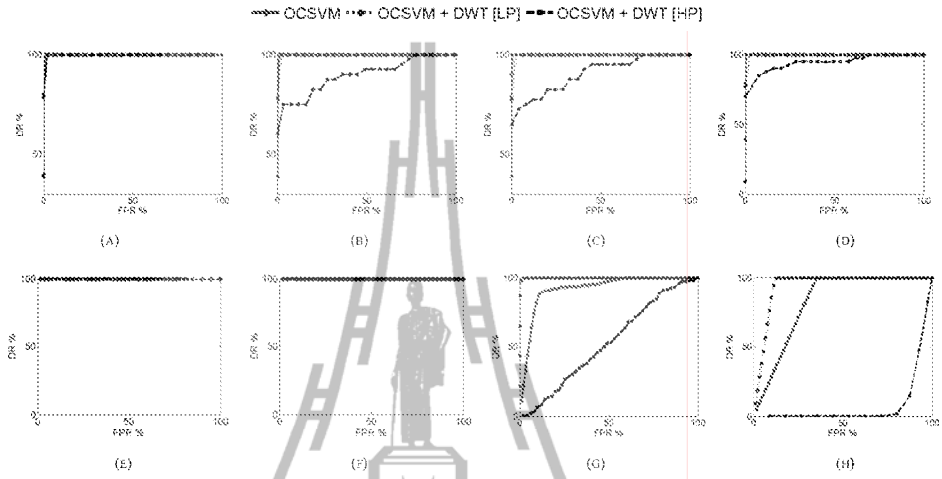


Figure 4 ROC curve for OCSVM using linear kernel (a) ROC for synthetic data inject 1/80 fault (b) ROC for synthetic data inject 5/16 fault (c) ROC for synthetic data inject 10/8 fault (d) ROC for synthetic data inject 20/4 fault (e) ROC for INTEL dataset (f) ROC for SS39 dataset (g) ROC for pdg2008-metro-1 dataset (h) ROC for NAMOS dataset.

C. *Comparison with previous work*

In this section, the best result of the proposed OCSVM+DWT algorithm from each dataset was selected to be compared with the results from the OCSVM alone, SOM alone and the SOM+DWT algorithms [3], [4] under the same datasets described previously. For each dataset, we selected the values of ν to use in our proposed algorithm which gave the best performance (the highest DR, the lowest FPR) as follows: ν = 0.06 for the all synthetic datasets, the pdg2008-metro-1 dataset, ν = 0.02 for the INTEL dataset and the SS39 dataset, ν = 0.22 for the NAMOS dataset. The SOM alone and SOM+DWT algorithms were trained with 50 iterations to prevent under-trained conditions using a 50 by 50 neuron grid. The number of samples used to train the SOM alone and the SOM+DWT algorithms were selected from a fault-free region using 2000 samples for INTEL and pdg2008-metro-1, 3000 samples for NAMOS datasets and 3200 samples for SS39 datasets.

Figure 5 (a) shows that using OCSVM alone and the OCSVM+DWT algorithms with synthetic data injected by 1/80 short faults obtained 100% DR, though OCSVM+DWT obtained marginal FPR of 2.12% and the OCSVM alone 0.99%. The SOM alone and SOM+DWT results were more conservative attaining less DR and no FPR. As the faults became more bursty (more noise faults) in Figures 5 (b), (c) and (d), 100% DR was obtained by OCSVM alone and OCSVM+DWT (LP) whereas OCSVM+DWT (HP) obtained 75-85% DR. However, this was at the expense of an increase in FPR of 2.12%, 7.68%, and 8.21% respectively for OCSVM alone, OCSVM+DWT (LP) and OCSVM+DWT (HP). Note that SOM alone with uncompressed data gave 86-90% DR with no FPR. SOM+DWT (LP) gave 76.4-83.7% DR with no FPR but using just half of the input data.

As for the real world datasets, INTEL and SS39 contained only short faults and were therefore easy to detect. Results in Figures 5 (e) and (f) agree with Figure 5 (a) with all OCSVM based algorithms obtaining 100% DR but FPR up to 1.9%. On the other hand, the SOM based algorithms obtained 100% DR but with FPR slightly lower of up to 1.09%. The improved performance for all algorithms was possibly due to the fact that the short faults in the INTEL and SS39 datasets were detected more easily than the synthetic dataset. Figure 5 (g) depicts results for the pdg2008 dataset which contained noise faults. Note that with the presence of noise faults, FPR for OCSVM based algorithms was greater than the SOM based algorithms which agreed with results in the synthetic dataset in Figures 5 (b), (c) and (d). However, OCSVM+DWT (LP) gave the best results obtaining 99.7% DR with 2.64% FPR, thereby outperforming OCSVM alone. Figure 5 (h) illustrates the results for the NAMOS dataset which comprised of constant faults. Such faults were difficult to detect since they appear as normal data. Even with SOM alone and SOM+DWT (LP) can fail to detect such faults if under-trained [3], [4]. Note that SOM alone, SOM+DWT (LP) and OCSVM+DWT (LP) all attained 100% DR, though the FPR was 12.77% for OCSVM+DWT (LP) but negligible FPR for SOM alone, SOM+DWT (LP). These results suggest that with data compression and using just half of the data input, OCSVM+DWT (LP) algorithm is suited for short and noise faults whereas SOM+DWT (LP) is suited for short and constant faults

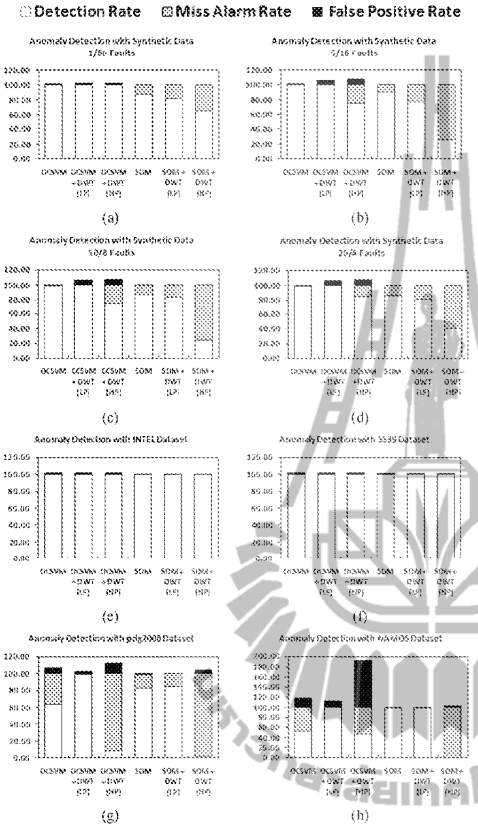


Figure.5 Detection Rate with different algorithm for (a) 1/80 fault synthetic data (b) 5/16 fault synthetic data(c) 10/8 fault synthetic data (d) 20/4 fault synthetic data (e) INTEL dataset (f) SS39 dataset (g) pdg2008-metro-1 dataset (h) NAMOS dataset.

V. CONCLUSION

We proposed the integration of OCSVM and DWT for anomaly detection in WSNs. We numerically evaluated the algorithm using MATLAB and tested it with both synthetic data and real world datasets. For synthetic data, our proposed algorithm with LP coefficients achieved 100% DR with marginal increase in FPR when compared with all other algorithms. For real world datasets, our proposed algorithm performed best by achieving nearly 100% DR although with slightly higher FPR for datasets containing short and noise faults. These results suggest that with data compression and using just half of the data input, OCSVM+DWT (LP) algorithm is suited for short and noise faults whereas SOM+DWT (LP) is suited for short and constant faults.

REFERENCES

[1] H.G. Goh, M.L. Sim, and H.T. Ewe, "Agriculture monitoring," in*Sensor Networks and Configuration*, Springer, 2007, pp. 439 - 462.

[2] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly Detection in Wireless Sensor Networks," in*IEEE Wireless Communications*, vol. 15, No. 4,Aug. 2008, pp. 34 - 40.

[3] S. Siripanadorn, W. Hattagam, and N. Teaumroong, "Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets." in*The 9th Int. Conf. on Applied Computer Science (WSEAS)*, Japan, Oct., 2010, pp. 291 - 297.

[4] S. Siripanadorn, W. Hattagam, and N. Teaumroong, "Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets." in*The Int. Journal of Communications*, Vol. 4, No. 3, Dec., 2010, pp. 74 - 83.

[5] F. Wang, Y. Qian, Y. Dai, and Z. Wang, "A Model Based on Hybrid Support Vector Machine and Self-Organizing Map for Anomaly Detection," in *Proc. 2010 Int. Confon Communications and Mobile Computing (CMC)*,Vol. 1, Apr 12-14, 2010, pp. 97 - 101.

[6] S. Rajasegarar, C. Leckie, M. Palaniswami, and J.C. Bezdek, "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks,"in *Proc. IEEE Int. Conf. on Communications 2007*,Jun 24 - 28, 2007, pp. 3864 - 3869.

[7] P. Laskov, C. Schafer, and I. Kotenko, "Intrusion detection in unlabeled data with quarter sphere support vector machines," in*Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, Vol. 27, No. 4, 2004, pp. 228 - 236.

[8] Y. Zhang, N. Meratnia, and P. Havinga, "Adaptive and Online One-Class Support Vector Machine-based Outlier Detection Techniques for Wireless Sensor Networks," in *Proc. 2009 Int. Conf. on Advanced Information Networking and Applications Workshops*, May 26 - 29, 2009, pp. 990 - 995.

[9] S. Rajasegarar, C. Leckie, J.C. Bezdek, and M. Palaniswami, "Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines for Anomaly Detection in Sensor Network," in *Proc. IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 3, Sep. 2010, pp. 518 - 533.

[10] D. M. J. Tax, and R. P. W. Duin, "Support vector data description," in*Machine Learning*, Vol. 54, No. 1, 2004, pp. 45 - 66.

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

[12] The Intel Lab. (2004). *INTEL dataset.*
[Online]. Available: http://berkeley.intel-research.net/labdata/

[13] The SensorScope Lausanne Urban Canopy Experiment (LUCE) Project. (2006). *SENSORSCOPE dataset.*
[Online]. Available: http://sensorscope.epfl.ch/index.php/LUCE.

[14] Network Aquatic Microbial Observing System. (2006). *NAMOSdata.*
[Online]. Available: http://robotics.usc.edu/~namos/data/jr_aug_06/

BIOGRAPHY

Ms.Saowaluk Takianngam was born on January 15, 1988 in Nakhonratchasima, Thailand. She finished high school education from Suranaree Wittaya School, Nakhonratchasima. In 2009, she received her Bachelor's Degree in Engineering (Computer) from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. For her post-graduate, she continued to study for her Master's degree in the Telecommunication Engineering Program, Institute of Engineering, Suranaree University of Technology.