OBSERVING THE UNOBSERVABLE

Distributed and Online Outlier Detection in Wireless Sensor Networks

Yang Zhang

Observing the Unobservable
Distributed Online Outlier Detection in Wireless Sensor Networks

Yang Zhang

Composition of the Graduation Committee:

Prof. Dr.	P.J.	Gellings	(UT, SECR)
Prof. Dr.	P.J.M.	Havinga	(UT, PS)
Dr.	N.	Meratnia	(UT, PS)
Prof. Dr.	P.M.G.	Apers	(UT, CTIT)
Prof. Dr.	J.L.	Hurink	(UT, DMMP)
Prof. Dr.	M.	van Steen	(VU Amsterdam)
Prof. Dr.	M.	Beigl	(TU Braunschweig, Germany)
Prof. Dr.	M.	Palaniswami	(University of Melbourne, Australia)

This research was conducted within the EU projects e-SENSE and SENSEI.

UNIVERSITY OF TWENTE.

Pervasive System Research Group
The Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente, The Netherlands.

CTIT

Center for Telematics and Information Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands.

Keywords: Wireless Sensor Networks, Outlier Detection, Distributed, Online.

Cover Design: Yang Zhang; Image by permission from http://www.sensorscope.ch/.

Printed by Wöhrmann Print Service.

ISBN 978-90-365-3058-3
CTIT PhD Thesis Series Number 10-174

OBSERVING THE UNOBSERVABLE
DISTRIBUTED ONLINE OUTLIER DETECTION IN WIRELESS SENSOR NETWORKS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the Rector Magnificus,
prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Wednesday the 23rd of June 2010 at 15:00

by

Yang Zhang

born on the 20th of January 1980

in Zhenjiang, China

This dissertation is approved by:

Prof. Dr.	Paul Havinga	(promotor)
Dr. Ir.	Nirvana Meratnia	(assistant promotor)

Abstract

The generation of wireless sensor networks (WSNs) makes human beings observe and reason about the physical environment better, easier, and faster. The wireless sensor nodes equipped with sensing, processing, wireless communication and actuation capabilities can be densely deployed in a wide geographical area and measure various parameters continuously from the physical world. Compared with traditional environmental sensing technologies, such densely deployed WSNs enable collection of fine-grained high spatial and temporal resolution data with less installation, maintenance, and operation costs.

However, raw sensor observations often have low data quality and reliability due to both internal and external factors including low quality of cheap sensors, dynamicity of network conditions, and harshness of the deployment environment. Use of low quality sensor data in any data analysis and decision making process will not only negatively impact analysis results and decisions made but also waste huge amount of valuable and limited network resources such as energy, as many incorrect values are transmitted. Low quality sensor data also prevents WSNs to fulfill their promises in terms of reliable real-time situation-awareness, as the low quality sensor data may generate large number of false alarms.

Motivated by the need to improve quality of data analysis and decision making, enhance efficiency of using WSNs resources by preventing unnecessary transmission of erroneous sensor observations, and increase effectiveness of monitoring and situation-awareness capabilities of the WSNs, in this thesis we focus on online identification of outliers whenever and wherever they occur. Outliers in WSNs are those observations that represent erroneous values (errors) or indicate particular phenomenal changes (events). Our outlier detection techniques, which are based on distributed in-network data processing, identify sensor observations that do not conform to normal behavior of sensor data without using a pre-defined threshold or triggering conditions.

Our main research objective is to design and implement effective and efficient outlier detection techniques for WSNs to identify outliers in an online and dis-

tributed manner and distinguish between errors and events with high accuracy and low false alarm, while maintaining the communication, computation and memory complexity low. Main contributions of this thesis can be summarized as:

1. **Taxonomy of and guideline for outlier detection techniques for WSNs**. We present shortcomings of existing outlier detection techniques and a set of important issues for outlier detection techniques for WSNs. We further provide a technique-based taxonomy to categorize current outlier detection techniques developed for WSNs and provide a guideline on requirements of suitable outlier detection techniques for WSNs.

2. **Design and comparison of data labelling techniques for performance evaluation of outlier detection techniques.** Many WSN applications suffer from lack of labelled data. To solve this problem, various labelling techniques are used offline to give semantic to data collected by WSNs and distinguish between normal data and outliers. We investigate impact of data distribution and data dependencies on four of these labelling techniques and evaluate their performance for the outlier detection process.

3. **Statistical-Based outlier detection techniques for WSNs.** We take two approaches in designing our outlier detection techniques. One approach originates from the field of statistics, while the other comes from the field of data mining and machine learning. Considering that spatio-temporal correlation exists between sensor observations, we use statistical approaches to quantify this correlation and to identify outliers in an online and distributed manner and distinguish between errors and events in real-time.

4. **Spherical support vector machine (SVM)-based outlier detection techniques for WSNs.** From data mining and machine learning perspective, we propose our distributed and online outlier detection techniques based on quarter-sphere one-class SVM. These techniques do not take into account correlation that may exist between data attributes. We simplify the process of modelling the quarter-sphere SVM to fit limited resources of WSNs and present three strategies to update the SVM-based model that represents normal behavior of sensor data.

5. **Ellipsoidal support vector machine (SVM)-based outlier detection techniques for WSNs.** We extend our quarter-sphere one-class SVM by taking into account correlation between different attributes to identify multivariate outliers. This results in our ellipsoidal SVM-based outlier detection techniques. To cope with dynamic nature of sensor data, we propose an efficient strategy to update the SVM normal model.

Samenvatting

De huidige generatie van draadloze sensornetwerken maakt het mogelijk de fysieke omgeving beter, gemakkelijker en sneller te observeren en te interpreteren. Een draadloos sensornetwerk bestaat uit draadloze sensornodes, die uitgerust zijn met sensoren, actuatoren, een microprocessor en draadloze communicatiemogelijkheden. Deze componenten kunnen in een groot geografisch gebied geplaatst worden, waar ze een netwerk vormen met een variabele dichtheid en waar ze voortdurend verschillende omgevingsvariabelen meten. Vergeleken met traditionele meetsystemen die ingezet werden om de omgeving te monitoren, maken deze draadloze sensornetwerken het mogelijk om gegevens met een hoge resolutie in tijd en ruimte te verzamelen, waarbij de installatie-, onderhouds- en gebruikskosten minder zijn dan bij de traditionele systemen.

Ruwe sensorgegevens zijn vaak van lage kwaliteit en hebben zijn vaak onbetrouwbaar door zowel interne als externe factoren, zoals de slechte kwaliteit van goedkope sensoren, de dynamiek van het netwerk en de soms barre omstandigheden waarin het netwerk zich bevindt. Wanneer sensorgegevens van slechte kwaliteit gebruikt worden voor data-analyse en als input voor beslissingsprocessen, zal dit niet alleen een negatieve invloed hebben op de resultaten van deze analyse en op de beslissingen die genomen worden, maar zal dit ook de beperkte middelen die een sensornode tot zijn beschikking heeft verspillen; zo kost het verzenden van nutteloze foutieve data energie – iets waar de node maar weinig van tot zijn beschikking heeft. Van draadloze sensornetwerken wordt verwacht dat ze betrouwbaar zijn, dat ze zich bewust zijn van hun omgeving, en dat ze direct (real time) reageren op gebeurtenissen in hun omgeving. Door het gebruik van sensoren van slechte kwaliteit kunnen deze verwachtingen niet waargemaakt worden: sensoren van slechte kwaliteit zullen bijvoorbeeld vaak een vals alarm veroorzaken.

Verschillende drijfveren liggen ten grondslag aan de focus op online identificatie van afwijkingen in de sensordata in dit proefschrift: de noodzaak tot het verbeteren van de data-analyse en de daaruit resulterende beslissingen, het ver-

hogen van de efficiëntie van het gebruik van de voor de node beschikbare bronnen, en het voorkomen van het verzenden van foutieve observaties van sensoren. Hierdoor zullen de nodes beter in staat zijn de omgeving te monitoren en zullen ze zich meer bewust zijn van hun omgeving. Afwijkingen of uitschieters in sensordata zijn observaties die aangeven dat ofwel foutieve waarden gelezen worden door een sensor, dan wel dat een bepaalde gebeurtenis optreedt in de omgeving van de sensor. Deze uitschieters in sensordata worden outliers genoemd. Onze detectietechnieken, die gebaseerd zijn op gedistribueerde in-netwerk dataverwerking, identificeren sensorobservaties waarvan de waarde afwijkt van de verwachte waarde, zonder gebruik te maken van voorgedefinieerde drempelwaarden of triggercondities.

De hoofddoelen binnen dit onderzoek zijn het ontwerpen en implementeren van effectieve en efficiënte outlierdetectietechnieken voor draadloze sensornetwerken, het online en op gedistribueerde wijze identificeren van outliers, en het maken van onderscheid tussen foutieve sensorwaarden en sensorwaarden die aangeven dat er een bepaalde gebeurtenis optreedt in de omgeving van de sensor. Hierbij streven we naar hoge nauwkeurigheid en een lage vals-alarmratio, waarbij de communicatie-, reken- en geheugencomplexiteit laag dienen te blijven. De hoofdbijdragen van dit proefschrift kunnen als volgt samengevat worden:

1. **Taxonomie van en richtlijnen voor outlierdetectietechnieken voor draadloze sensornetwerken.** We presenteren tekortkomingen van bestaande outlierdetectietechnieken en geven een overzicht van belangrijke problemen van deze technieken voor draadloze sensornetwerken. We presenteren een speciaal voor draadloze sensornetwerken ontwikkelde taxonomie om de huidige outlierdetectietechnieken te categoriseren en geven richtlijnen voor vereisten voor geschikte outlierdetectietechnieken voor draadloze sensornetwerken.

2. **Ontwerp en vergelijking van datalabeltechnieken voor het beoordelen van de prestaties van outlierdetectietechnieken.** Het beoordelen van de prestaties van outlierdetectietechnieken voor draadloze sensornetwerken wordt bemoeilijkt door het gebrek aan gelabelde data. Om dit probleem op te lossen, worden offline verschillende labeltechnieken gebruikt om betekenis te geven aan de gegevens die door draadloze sensornetwerken verzameld zijn en onderscheid te maken tussen normale data en outliers. We onderzoeken de invloed van de datadistributie en data-afhankelijkheid van vier van deze labeltechnieken en evalueren hun prestaties in relatie tot het outlierdetectieproces.

3. **Statistische outlierdetectietechnieken voor draadloze sensornet-**

werken. We nemen twee benaderingen in het ontwerpen van onze outlierdetectietechnieken. Eén benadering komt uit de statistiek, terwijl de andere uit het vakgebied van de datamining en machine learning komt. In overweging nemend dat de observaties van de sensoren in tijd en ruimte gecorreleerd zijn, gebruiken we verschillende statistische benaderingen om deze correlatie te kwantificeren en afwijkingen te identificeren op online en gedistribueerde wijze en maken we real time onderscheid tussen fouten en gebeurtenissen.

4. **Op spherical support vector machine(SVM)-gebaseerde outlierdetectietechnieken voor draadloze sensornetwerken.** We introduceren onze gedistribueerde en online outlierdetectietechnieken, gebaseerd op de quarter-sphere one-class SVM die afkomstig is uit het vakgebied van dataminingen machine learning. Deze techniek laat de correlatie die tussen de verschillende omgevingsvariabelen zou kunnen bestaan buiten beschouwing. We vereenvoudigen het modelleringsproces van de quarter-sphere SVM om te voldoen aan de gelimiteerde mogelijkheden van de draadloze sensornetwerken. We presenteren drie strategieën om het op SVM gebaseerde model dat het normale gedrag van de sensordata representeert aan te passen aan veranderingen in de omgeving.

5. **Op ellipsoidal support vector machine(SVM)-gebaseerde outlierdetectietechnieken voor draadloze sensornetwerken.** We breiden onze quarter-sphere one-class SVM uit door de correlatie tussen de verschillende omgevingsvariabelen mee te wegen, waardoor ook meerdimensionale outliers geïdentificeerd kunnen worden. Dit resulteert in onze op de ellipsoidal SVM gebaseerde outlierdetectietechniek. Om om te kunnen gaan met het dynamische karakter van de sensordata, stellen we een efficiënte aanpak voor om het model van de data dat de SVM gebruikt aan te passen aan verandering in de omgeving.

Acknowledgements

I have had a dream that I want to be a PhD and obtain the degree of doctor since I was very young. This dream may mainly result from the environment I grew up; I had been living at the campus of a university in China and my parents are both staff of the university. As time goes by, I have successfully finished four-year PhD research work and this thesis, and I am now actively preparing for my final public defence. Recalling from having a childhood dream to now infinitely approaching to it, except for my own unceasing endeavor, I certainly can not do without all those people who have ever helped and supported me, especially during my four-year PhD life in the Netherlands. Herewith, I would like to express my sincere appreciation to them for what they have done for me over these years.

First of all, I would like to express my deepest gratitude and respect to the two people who are the most important to me in the Netherlands: my promoter Paul Havinga and my daily supervisor Nirvana Meratnia. I feel all the time that I am so fortunate to meet both of them in my life. Paul was my master promoter. When I was wondering what I would do for my master project, it was Paul who brought me into the field of wireless sensor networks; I first got aware of what a WSN was and how sensor nodes worked together. Moreover, I was given the chance to involve in designing WSN protocols and further deploying and implementing sensor nodes prototypes in the real environment for testing. This great experience developed my capability of doing research and helped me integrate theoretical knowledge with practical experiments. Afterwards, Paul offered me this valued opportunity to be a PhD at today's Pervasive Systems group. I still remember that day was May 12, 2006.

Paul, as my PhD promoter, played a crucial role in guiding my research work towards the right direction. Although his schedule is extremely busy all the time, Paul always could make his appointments for me. Every time after I discussed with him, I always could obtain good advice from him. He also provided quite a lot chances for me so that I have worked for two EU projects e-SENSE and SENSEI, attended many high-quality conferences held all over the world, participated in

interesting summer schools, and even joined useful English courses on improving presentation and writing skills. Paul is an open-mind professor with his nice smile. His optimism and positive attitudes always encouraged me when I was struggling with difficulties in my work, especially during my thesis writing. For his understanding, encouragement, critical but helpful comments, my thesis can be successfully approved and my defence can be held as schedule. Furthermore, I will be very proud that I will become the first student he supervised as both master and PhD promoter if I get through my defence at that day.

Nirvana, I am so lucky to have her as my daily supervisor during the past four years. In my mind, she is a very friendly, enthusiastic, capable, patient and considerate supervisor. She has helped me everywhere. She always discussed with me about each of specific research questions, and provided me with her wonderful ideas and helpful suggestions. She often illustrated her ideas on the whiteboard and wrote down detailed comments on my notepaper. Moreover, she actively took part in my experiments on simulation and helped me improve experimental results. My all publications had benefited so much from her careful review over again so that I have no any reject record for all submissions. She also arranged the opportunity of internship for me to extend my research work at ITC, where I learned so much about geostatistics. Every time when I had some personal trouble or got depressed on my research work, she always smiled to encourage me, support me and help me. She has never doubt whether I could finish my PhD research and obtain the degree of doctor. In the third year of my PhD, when I heard she was promoted to assistant professor, I was extremely happy that she could still be my daily supervisor. Furthermore, I was really moved by her during my thesis writing. To help me catch the schedule, she ever thoroughly reviewed my thesis for several days at the office until deep night. Without her continuous support, encouragement and help, my thesis would not have been possible. As her first supervised PhD student, I really appreciate everything that Nirvana has done for me all these years.

In the last year of my PhD studies, I had the opportunity of internship to work at professor Alfred Stein's research group, Earth Observation Science of ITC. I deeply thank to him for accepting me as intern at ITC. Alfred is very professional at spatial and spatio-temporal statistics. He always gave me wonderful ideas during our discussion. Moreover, He reviewed my reports over again and provided many constructive comments to me. He also recommended me with some good references, from which I learned so much about geostatistics. This helped me extend my research work and finish Chapter 4 of my thesis. The other person I would like to express my sincere appreciation is Dr. Nicholas Hamm. He is my daily supervisor at ITC. I am greatly impressed by his rigorous scientific attitude. He always guided me how to do research in a more serious way, including how

to critically select high-quality publications as reference. Moreover, he is very patient and always would like to discuss any of research questions with me, even helping to check my programming code. Nick is also a good teacher. I like his course of Geostatistics very much. Furthermore, I am thankful to him for spending lots of time modifying and improving our journal paper.

Many thanks to all members of my graduation committee for reading my thesis: Prof. Paul Gellings, Prof. Peter Apers, Prof. Johann Hurink, Prof. Maarten van Steen, Prof. Michael Beigl, Prof. Marimuthu Palaniswami. Theirs insightful comments helped me improve the quality of my thesis and express my ideas better.

During these years, I have worked with a group of nice colleagues. They have created a friendly, helpful and interactive environment. I would like to thank all members and ex-members of Pervasive Systems group. I thank my roommate Aysegul for having a good time working together. We both always shared important and interesting information and told our feelings with each other. She always brought special Turkish food and gifts to me every time when she came back from Turkey. I enjoyed them very much. I thank my two paranymphs: Majid and Marlies for my support. Majid and I have lots of commons, although we come from different countries. We both are working at the same research group (PS), supervised by the same promoter Paul and daily supervisor Nirvana, and living in the same building (Matenweg 73) of the campus, even we both can speak a litter mother language of each other. I always would like to talk with him about any interesting issue. I thank him for bringing lots of fun and help to me. Marlies had done a key contribution for Chapter 3 of my thesis. She made numerous experiments and figures for labelling data, and provided required labelled data for me to evaluate the performance of my outlier detection techniques. The task of labelling data is very hard, but she always patiently worked with it and modified settings according to my requests. She also wrote the Dutch abstract for my thesis. I really appreciate all her contributions for my thesis.

I would like to specially thank Supriyo for give me opportunities to involve in the implementation of his protocols. From there, I had got lots of practical experience of WSNs. Moreover, I had learned so much from him how to do research when he was my roommate. I thank Mihai and Raluca for having a good memory playing badminton together and traveling during conferences. I had lots of fun to play with them. I thank Berend Jan and Arta for contributing my thesis. I often got helpful suggestions from them when I met some problems during working on my thesis. I thank Arie for implementing my outlier detection techniques in sensor nodes. I also would like to thank all ex-members of PS group: Jian, Kavitha, Ozlem, Stefan, Lodewijk. They are all good examples for me to be a qualified PhD, especially Jian. As my master supervisor, he always helped me

explain my doubts and questions about his algorithm, and also worked together with me in the whole course of implementation. Moreover, he encouraged me to apply for the PhD and shared his experience on studying and living abroad. I thank Rajasegarar for sharing his experimental dataset and implementation source code to evaluate my outlier detection techniques. I thank Tjerk and Leon of Ambient Systems for providing technical support for my implementation. I of course thank our nice secretaries, Nicole, Marlous, Thelma for making our life easier with their great administrative support.

At the beginning of my first master study in China, I made the important decision in my life that I left to the Netherlands for the new master. I did not expect at all that I have already lived in the Enschede for nearly seven years. It was very difficult for me from never leaving parents to living alone abroad at the beginning. Fortunately, I had known so many Chinese friends here, who make my life in Enschede more colorful. They are Xu Qi, Zhang Yelei, Chang Haiyue, Tian Jian, Liu Puming during my master life, Zhou Wei, Bai Wei, Zhang Qiwei, Wang Xinhui, Yang Jing, Zhao Yiping, Lao Jin & Shui Lingling, Cheng Wei, Ru Zhiyu, Zhou Wei & Zhao Wei, Tan Lianghui, Yang Di, Liu Chanjuan, Sheng Xiaoqing, Shao Xiaoying, Song Jing, Guo Rui, Song Chunlin, Wu Zhongkai, Li Rongmei, Li Yixuan, Xiao Li, Wang Xin, Xu Genjiu, and all the people who I have known in the Netherlands. I had so much fun with them. Although some of them are not in Enschede anymore, I will never forget those happy time with them.

The most important people in my life I should appreciate are my girlfriend Ge Rui and my parents. Thanking for the God's arrangement, I met Rui in the first year of my PhD. We had gone through the difficult separated life, and now we are getting together in the beautiful Netherlands. I deeply appreciate her attentive care and support for me, especially during the most difficult time of my thesis writing. She always encouraged me and gave me much more confidence and power to solve problems. I am very lucky to have her accompanying with me and giving me unlimited energy to move forward.

Last, I express my deepest appreciation and respect to my parents for understanding me, encouraging me and helping me all the time. Any accomplishment I ever made is contributed by their guidance. They will be very proud of me for making their and my dream come true. Therefore, I dedicate this work to my parents for their love and support.

Yang Zhang
June 2010
Enschede, The Netherlands

Table of Contents

Chapter 1

Introduction

Human beings have never lost the enthusiasm in discovering and understanding their world. In the course of history, people have carried out millions of observations and experimental work on the physical environment by collecting environmental data, analyzing and reasoning about it, and making sense of nature. This has enabled people to be more acquainted with their surroundings and better understand the previously "mysterious" nature and have it more under control.

With the increasing advances of science and technology in particular in the field of micro-electro-mechanical system (MEMS) technologies, wireless communications, and digital electronics, especially, in the past decade a new breed of tiny embedded systems known as *wireless sensor nodes* has emerged. This type of wireless sensor nodes are equipped with sensing, processing, wireless communication, and more recently actuation capability. A wide variety of sensors include temperature, humidity, sound, pressure, light, vibration, motion [4]. Figure 1.1 illustrates an example of a WSN sensor node. These sensor devices are capable of collaborating with each other in a self-organized ad-hoc manner to observe, process, and reason about the phenomena being monitored. A large collection of these devices forms a *wireless sensor network* (WSN) [2]. The generation of wireless sensor networks (WSNs) makes human beings observe and reason about the physical environment better, easier, and faster. These wireless sensor nodes can be densely deployed in a wide geographical area and measure various parameters continuously from the physical world. They are also able to perform limited local data processing and transmit raw or processed data via a single or multi hop routing to a *central station* (known as a gateway). Data from the gateway can be further accessed by people via wired or wireless networks.

Figure 1.1: The μNode from Ambient Systems [1]

Currently a diverse set of applications for WSNs cover different fields of personal, industrial, business, and military domains. Various applications of WSNs include environmental and habit monitoring, localization and target tracking, supply chain management, logistics and transportation, health and medical care, industrial monitoring and control, and battlefield observation [38, 4]. The newly emerged concept of *Internet of things* [52] seems to be the future of the WSNs. The Internet of things concept envisions every object to be equipped with sensor nodes and millions of these intelligent objects communicate with each other and constitute a network. Using this network of intelligent objects, human beings can easily and quickly know the state of objects and manage and control their environment remotely. As a paradigm shift from personal computing to *ubiquitous computing* [130], WSN is bringing the flexibility of information technology in every aspect of people daily life.

As an interdisciplinary field, WSN runs across many knowledge disciplines including signal processing, networking and protocols, embedded systems, information management and distributed algorithms. The broad spectrum of WSNs research includes MAC, routing, transport communication protocols, localization, time synchronization, query processing, scheduling, clustering, detection, classification, hardware design, operating systems, simulation tools, security and privacy [2, 123]. These various research issues all aim at enhancing the effectiveness and efficiency of WSNs applications in real-life.

1.1 Motivation of Outlier Detection in WSNs

The ultimate goal of the wireless sensor networks goes beyond monitoring and data collection. It concerns with timely data analysis and assessment and (near)

real-time, efficient, and accurate critical decision making and situation awareness. Any data analysis and decision making process relies heavily on amount and quality of data being processed as well as additional information and context.

1.1.1 Sensor Data Quality

It can be said that one of the success factors of WSN is quality of its observations, i.e., whether collected sensor data actually reflects *true state* of monitoring phenomenon. However, raw sensor observations collected from distributed deployed sensor nodes often are inaccurate and incomplete. These inaccurate and incomplete data may be in form of noise, missing values, duplicated or inconsistent data [45]. Very frequent occurrence of these observations have considerable negative impact on quality and reliability of sensor data. The main reasons of producing low quality and low reliable sensor data are:

- **Internal factors**. The internal factors come mainly from WSNs and sensor nodes themselves. Firstly, although sensor nodes are equipped with sensing, processing, wireless communication, and even actuation capability, the design purpose of sensor nodes is to be cheap and miniature, which stem from the fact that they need to be pervasively and in very large quantities deployed. This design purpose results in inherent resource constraints and limited capability of sensor nodes. More specifically, the wireless radio transceiver used in WSNs has a low data rate (typically between 10 and 100 kbps) and low coverage (typically between 20 and 200 m) [77]. The microcontroller processing unit used in WSNs is usually associated with limited computational power (typically 8 or 16 bit CPUs at 4-8 MHz), while the storage space is in the order of 10kB random access memory (RAM) and 48kB programmable flash memory [78]. This relatively low-capability hardware greatly influences the quality of collected sensor data. Despite the fact that the sensor platform technology is becoming enhanced, the chance of generating inaccurate and incomplete data is still quite high in real-life. Secondly, wireless sensor nodes are usually battery-powered. Due to the fact that batteries used for sensor nodes (typically with approximately 1500 mAh) [24] do not last long, by decrease of nodes' battery level the probability of generating incorrect data is growing rapidly [104]. Thirdly, sensor data may be impacted by dynamic nature of communication link as well as the network topology due to addition or failure of sensor nodes. The large scale and high density of the wireless sensor network and mobility of the nodes may also influence data quality [143].
- **External factors**. The external factors are mainly originated from mon-

itoring environment and human beings. On the one hand, WSNs may be deployed to operate in harsh and unattended environments, e.g., mountains, forests, rivers, deserts. The harshness of the deployment area may make sensor data more vulnerable to generate erroneous or spurious data [7]. On the other hand, WSNs may suffer from human factors like malicious attacks. Sensor nodes may be deployed in restricted areas of adversaries, in which data generation and processing would be attacked by adversaries. Denial of service attacks, black hole attacks and eavesdropping [86] are examples of these attacks. Human-related factors may also be in the form of accidental move or destruction of the sensor nodes [3].

All these internal and external factors cause low quality and unreliable sensor data. The generated low quality sensor data has various impacts. Firstly, it seriously impacts the effectiveness of monitoring capability of the WSNs to an extend that people may fail to understand the environment well. Secondly, transmission of low quality data results in huge waste of valuable and limited network resources. Thirdly, real-time decision making and situation awareness capability of the WSNs will be hugely influenced. Even worse, the inaccurate and unreliable sensor data may increase generation of false alarms and erroneous decisions.

Many WSNs research efforts have been directed on node platform design and protocol optimization to save the sensor node energy and enhance the efficiency of WSNs resource usage, while little attention has been paid to the quality of sensor data itself. With more deployments of real sensor networks [111, 41, 51, 108], in which the main function is to collect interesting data and to make intelligent decisions, improving quality and reliability of sensor data is becoming a crucial step in order to make WSNs an ideal sensing and actuation tool.

1.1.2 Outlier Detection in WSNs

One solution to ensure quality of sensor data is through online detection of outliers whenever and wherever they occur. The term *outlier* originally stems from the statistics community [31]. Coming across various definitions of an outlier, it seems that no universally accepted definition exists. The notion of outliers has been shown to differ in terms of specific application domains, data types and utilized detection techniques [144]. Two classical definitions of outliers are provided by Hawkins [43] and Barnett and Lewis [10]. The former defines, "an outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism", where as the latter defines "an outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data". The two above definitions

imply that an outlier in a dataset is an observation that is significantly different from the majority of observations in the dataset.

In the context of WSNs, outliers, also known as *anomalies*, can be abstractly defined as *those sensor observations that do not conform to the defined (expected) normal behavior of sensor data* are considered as *outliers*. Based on this definition, in this thesis, we classify outliers occurred in WSNs into the following:

- **Errors**. This sort of outliers refers to those sensor observations that do not reflect the true state of monitoring phenomena and significantly deviate from a priori defined normal behavior of the data. Due to the fact that these erroneous observations seriously influence the quality of data analysis, they need to be removed from the dataset or corrected if possible. One should note that outliers caused by malicious attacks also belong to errors as these observations do not reflect the true state of monitoring things.

- **Events**. This sort of outliers refers to those sensor observations that do not conform to a priori defined normal behavior of the data but reflect the changing state of monitoring phenomena. These outliers may indicate particular gradual state changes of the real-world, e.g., a global environmental change (climate warming and cooling) or an unexpected sudden event (forest fire, earthquake, chemical spill, and air pollution). Compared to errors, investigating these outliers can deepen the understanding about the environment.

Outlier detection differs from *fault detection*, *event detection*, and *intrusion detection* in a sense that fault detection is specifically targeted to identify erroneous sensor data using given thresholds [17, 67, 84, 20]. Event detection is specifically targeted to identify specific events using the trigger condition or semantic of these events [70, 25, 60, 127, 109, 91, 142]. Intrusion detection is specifically targeted to identify potential malicious attacks from the security perspective [110, 11, 85, 69]. Our focus on outlier detection, however, is on the process of outlier detection in WSNs through using data analysis. Our outlier detection aims to identify anomalous observations that do not conform to defined normal behavior of sensor data, without any given threshold or trigger conditions. By doing this, we can (i) better understand sensor data characteristics and its internal structure, (ii) more accurately define the normal behavior of sensor data, (iii) identify the intrinsic change of normal behavior of sensor data and unexpected phenomena, (iv) identifying erroneous sensor data (malicious attacks concerned with the issue of network security is out of the scope of this thesis). Use of outlier detection in WSNs will improve robustness of data analysis, enhance the efficiency of using WSNs re-

sources, avoid the unnecessary transmission of erroneous sensor observations and reduce energy consumption, eventually ensure the effectiveness of using WSNs.

In what follows, we further exemplify the essence of outlier detection through several real-life applications and show how outlier detection is a quite critical part of them.

- **Environmental monitoring**, in which sensors are deployed in harsh and unattended regions to monitor the natural environment. Outlier detection can identify when and where unusual event occurs and trigger an alarm upon detection. For instance, chemical sensors gathering the chemical data are used to monitor toxic spills and nuclear incidents.

- **Habitat monitoring**, in which endangered species can be equipped with small non-intrusive sensors to monitor their behavior. Outlier detection can indicate abnormal behavior of the species and provide a closer observation about behavior of individual and groups of animals.

- **Health and medical monitoring**, in which patients are equipped with small sensors on multiple positions of their body to monitor their well-being. Outlier detection showing unusual records can indicate whether the patient has potential health problems and allow doctors to take effective medical measures in time.

- **Industrial monitoring**, in which machines are equipped with temperature, pressure, or vibration amplitude sensors to monitor their operation. Outlier detection can quickly identify anomalous readings to indicate possible malfunction or any other abnormality in the machines and allow for their corrections.

- **Localization and tracking**, in which sensors are embedded in moving targets to track them. Outlier detection can filter erroneous information in raw data to improve the estimation of the location of targets and also to make tracking more efficiently and accurately.

- **Surveillance monitoring**, in which multiple sensitive and unobtrusive sensors are deployed in restricted areas. Outlier detection identifying the position of the source of the anomaly can prevent unauthorized access and potential attacks by adversaries in order to enhance the security of these areas.

1.2 Research Objectives

As mentioned before, outlier detection is rather crucial for WSNs. Therefore, in this thesis, our main research objective is to design and implement effective and efficient outlier detection techniques for WSNs. In order to better specify our objectives, we first need to elaborate on what effectiveness and efficiency mean in this context:

- **Effectiveness**. This sub-objective is concerned with the accuracy of detecting outliers from normal observations in WSNs. The detection accuracy can be evaluated by two performance metrics, i.e., *detection rate (DR)* and *false alarm rate*, also known as *false positive rate (FPR)*. The detection rate represents the percentage of outliers that are correctly identified. The false alarm rate represents the percentage of normal observations that are incorrectly considered as outliers. An effective outlier detection technique is required to maintain a high detection rate while keeping the false alarm rate low. Since, outliers occurred in WSNs may be type of error or event, an effective outlier detection technique for WSNs needs to correctly distinguish between the two, and deal appropriately with them.

- **Efficiency**. This sub-objective is concerned with the efficient use of WSNs resource. As we described, size and cost constraints on sensor nodes result in severe resource constraints. In WSNs, the scarcest and most crucial resource is *energy*. Data transmission is the main source of energy consumption in the network [92]. Thus an efficient outlier detection technique for WSNs needs to have low communication overhead. This requirement may be fulfilled when outlier detection in WSN is performed in a *distributed* manner instead of a *centralized* manner. Traditional centralized manner of continuously transmitting sensor observations from sensor nodes to the central station for data analysis causes large amount of communication overhead as well as high energy and bandwidth consumption. It also does not scale well when the size of the network increases. On the contrary, distributed processing (partial or complete) of data locally on sensor nodes reduces the transmission of raw sensor observations and makes efficient use of energy and bandwidth. Moreover, other resources such as computational power and memory space are also limited for sensor nodes. Thus an efficient outlier detection technique needs to have low computational and memory complexity so that it can quickly detect outliers especially in case of detecting events. This requirement may be fulfilled when outlier detection in WSN is performed in an *online* manner instead of an *offline* manner.

According to the above elaboration, we can further specify our research objective in this thesis as:

To design and implement effective and efficient outlier detection techniques for WSNs to identify outliers in an online and distributed manner and distinguish between errors and events with high accuracy and low false alarm, while maintaining the communication, computation and memory complexity low.

We are well aware of the trade-off between effectiveness and efficiency. After careful analysis of this trade-off, we take this trade-off into account during the design of our outlier detection techniques.

1.3 Thesis Contributions

To achieve our research objective, we describe in the following the main contributions of this thesis.

Contribution 1: Taxonomy and guideline of outlier detection techniques for WSNs.

Since no taxonomy and guideline for outlier detection techniques specifically developed for WSNs exists, we first summarize several important issues for outlier detection techniques for WSNs. We then introduce general outlier detection techniques based on various types of techniques as well as based on the degree of using pre-labelled data and highlight their shortcomings that make them not directly applicable for WSNs. We further provide a technique-based taxonomy to categorize current outlier detection techniques specifically developed for WSNs and present an extensive overview of these techniques. By comparing these techniques against requirements and challenges faced in WSNs, we provide a guideline on requirements that suitable outlier detection techniques for WSNs should meet. This provided guideline will be considered as design criterion and performance metrics for our proposed outlier detection techniques for WSNs in later chapters.

Contribution 2: Design and comparison of data labelling techniques for performance evaluation of outlier detection techniques for WSNs.

Since sensor data is unlabelled and since no general purpose labelling techniques exist for outliers, we investigate and compare four data labelling techniques based on Mahalanobis distance, density, running average, and Bayesian networks for identification of various types of outliers occurring in a real environmental dataset. After describing fundamentals of these techniques, we present a thorough comparison between these labelling techniques based on the real dataset in

terms of performance and complexity and the effect of the data characteristics on the labelling. Experiments results indicate that the choice of the labelling techniques is very important and has great impact on performance evaluation of outlier detection techniques. Furthermore, we choose three labeling techniques to label data for evaluation of our proposed outlier detection techniques in later chapters.

Contribution 3: Statistical-Based outlier detection techniques for WSNs.

Considering that sensor data collected from densely deployed sensor nodes in the physical environment tends to be correlated in space and time, we efficiently quantify spatial and temporal correlations of sensor data and exploit them to propose distributed and online outlier detection techniques for WSNs based on temporal correlation, spatial correlation and spatio-temporal correlations. These proposed techniques enable each node to identify outliers and distinguish between errors and events in real-time. Specifically, we utilize time series analysis to obtain temporal correlation and use geostatistical data analysis to obtain spatial correlation. Moreover, we take into account the efficiency of our outlier detection techniques, which are designed to reduce computational and memory complexity, and minimize consumption of communication in WSNs. Experimental results reveal that taking spatio-temporal correlations into account in outlier detection techniques, contributes to thorough understanding of the internal structure of sensor data, and precise identification of outliers and detection of the change of normal behavior in WSNs.

Contribution 4: Spherical support vector machine (SVM)-based outlier detection techniques for WSNs.

To avoid the assumption on explicit probability distribution, we propose distributed and online outlier detection techniques for WSNs based on quarter-sphere one-class SVM originated from the data mining and machine learning community. We first simplify the process of modelling the quarter-sphere SVM to meet requirements of the WSNs and efficiently utilize it to identify outliers in multivariate sensor data. We also take advantage of the theory of spatio-temporal correlations to precisely detect outliers and the change of normal behavior of sensor data. Furthermore, we present three strategies to update the SVM model that represents normal behavior of sensor data in order to cope with dynamic nature of sensor data. Experimental results show that our proposed outlier detection techniques have the ability to precisely detect outliers and the change of normal behavior in sensor data streams and are robust in terms of parameter selection.

Contribution 5: Ellipsoidal support vector machine (SVM)-based outlier detection techniques for WSNs.

We extend our quarter-sphere one-class SVM by taking into account correlation between different attributes to identify multivariate outliers. This results in our ellipsoidal SVM-based outlier detection techniques. We simplify the process of modelling the hyperellipsoidal SVM to fulfill requirements of the WSNs and efficiently utilize it to identify multivariate outliers. To cope with dynamic nature of sensor data, we propose an efficient strategy to update the SVM normal model. Experimental results show that compared to previously proposed spherical SVM-based outlier detection techniques, our ellipsoidal SVM-based outlier detection techniques achieve better detection accuracy and lower false alarm.

1.4 Thesis Organization

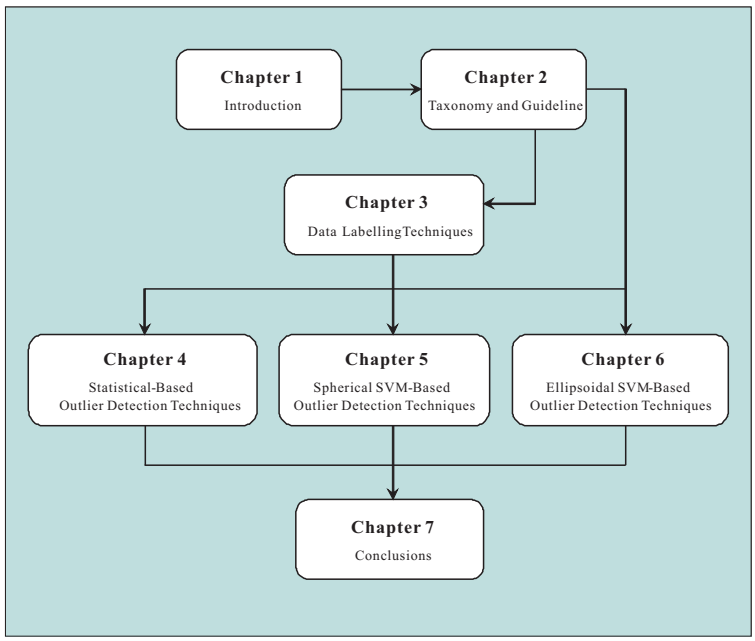


Figure 1.2: Organization of the thesis

The organization of the thesis is shown in Figure 1.2, which shows the information flow of the main research directions, the contributions, and relationship among different thesis chapters. Chapter 2 provides a technique-based taxonomy of current state-of-the-art on outlier detection techniques that are used in WSNs,

and also provides a guideline for selecting suitable outlier detection techniques for WSNs. It lays the foundations for the work presented in Chapters 3-6. Chapter 3 investigates and compares different data labelling techniques. Results of this chapter are used for evaluation of our proposed outlier detection techniques presented in Chapter 4-6. Chapter 4 proposes spatio-temporal correlations-based outlier detection techniques from the statistical perspective, while Chapter 5 and 6 propose spherical and ellipsoidal SVM-based outlier detection techniques from the data mining and machine learning perspectives. We conclude this thesis in Chapter 7 by summarizing the key results and highlighting the open research areas that still need to be investigated.

Chapter 2

Taxonomy and Guideline of Outlier Detection Techniques for Wireless Sensor Networks

To ensure high data quality, secure monitoring, and reliable detection of interesting and critical events in wireless sensor networks outlier detection mechanisms need to be in place. Outlier detection techniques can be categorized based on their application domains, data types they deal with, and fields of research they originate from. Outlier detection is not a new topic. However, in this chapter, we explain why traditional outlier detection techniques do not suffice for WSNs. Before doing so, we first identify several important issues that need to be considered when dealing with outlier detection in WSNs and then provide a technique-based taxonomy to categorize current state-of-the-art on outlier detection techniques specifically developed for WSNs. In addition to presenting an extensive overview of these techniques, we also compare them and provide a guideline on requirements that suitable outlier detection techniques for WSNs should meet.

2.1 Introduction

Outlier detection, also known as *anomaly detection*, *deviation detection* or *novelty detection*, is one of the fundamental tasks of *data mining* along with predictive modelings, cluster analysis, and association analysis [119]. Compared with these other three tasks, outlier detection is the closest to the initial motivation behind data mining, i.e., *discovering hidden interesting information from large databases* [45]. Existing outlier detection techniques can be categorized depending on several different principles. For instance, they can be categorized based on their *application domains*, the *data types* they deal with, and the *fields of research* they originate from. Based on application domains, current outlier detection techniques can be classified into cyber-intrusion detection, fault detection, medical and health detection, industrial damage detection, image processing detection, textual detection, and also sensor network [18]. Outlier detection in these applications aim at identifying instances of *unusual activities*. Based on the data types they deal with and data characteristics, outlier detection techniques can be categorized into techniques dealing with simple data, high dimensional data, mixed-type attributes data, sequence data, spatial data, streaming data, spatio-temporal data [144]. Classification based on the fields of research outlier detection techniques originate from results in categorizing these techniques into statistic, data mining, machine leaning, information theory, and spectral decomposition based approaches [18].

In line with these classifications, Markos and Singh [71, 72] present an extensive review of outlier detection techniques based on statistical and neural network approaches. Hodge and Austin [47] address outlier detection techniques developed based on statistics, neural networks, and machine learning approaches. Chandola et al. [18] classify outlier detection techniques in terms of diverse application domains and research areas. Zhang et al. [144] provide a taxonomy for outlier detection techniques with respect to multiple types of datasets. Despite all these efforts, none of these taxonomies address outlier detection techniques specifically developed for WSNs. Moreover, there is no guideline on requirements that suitable outlier detection techniques for WSNs should meet.

Therefore, we in this chapter present several issues that need to be considered when dealing with outlier detection in WSNs in Section 2.2. General outlier detection techniques and their shortcomings that make them not directly applicable for WSNs are highlighted in Section 2.3. A technique-based taxonomy to categorize current state-of-the-art on outlier detection techniques specifically developed for WSNs and their overview will be presented in Section 2.4. A comparison between these techniques and a guideline on requirements for suitable outlier detections techniques for WSNs are presented in Section 2.5. Finally this chapter is

concluded in Section 2.6 and the provided guideline will be considered as design criterion and performance metrics of our outlier detection techniques proposed in the following chapters.

2.2 Important Considerations for Outlier Detection Techniques for WSNs

Accuracy and execution time of outlier detection techniques vary for different application domains and data characteristics. This implies that no single universally applicable or generic outlier detection technique exists [47]. Thus, designing an appropriate outlier detection technique for WSNs is important. In this section, we identify several important issues that need to be considered when dealing with outlier detection in WSNs.

2.2.1 Sensor Data Characteristics

Sensor data collected by WSNs has its unique characteristics, which should be taken into account while designing outlier detection techniques to ensure their performance. Typical characteristics of sensor data in WSNs are:

- **Streaming data**. Sensor data is intrinsically streaming data, which means that a large volume of data is continuously collected by sensor nodes [36]. Frequent change of streaming data may change the normal behavior of sensor data over time. Implication of this for outlier detection is that a priori defined normal behavior of sensor data may not be sufficiently representative in the future.

- **Continuous attributes**. While sensor data has *continuous attributes* [118], it does not have any categorical [118] or mixed-type attributes. This implies that the values of sensor data are all real values, e.g., temperature, height, or weight.

- **Univariate & Multivariate attributes**. Sensor data may consist of only one attribute (*univariate*) or multiple attributes (*multivariate*). Generally, sensor data has low-dimensional attributes due to resource limitation of sensor nodes. *Univariate outlier* represents a single attribute detected as outlier, while *multivariate outlier* represents a combination of multiple attributes showing anomalous values, even if none of the attributes individually is detected as outlier [103].

- **Distributed data**. Due to the fact that sensor nodes are distributedly deployed, each sensor node has a limited knowledge about the monitoring phenomena, which may not correctly and completely represent the normal behavior of sensor data.
- **Unlabelled data**. For sensor data often no pre-labelling is available to define normal behavior of sensor data or evaluate performance of outlier detection techniques.
- **Data correlations**. Two types of correlations may exist in sensor data, i.e., (i) correlation between data attributes, and (ii) correlation between sensor node's own observations and observations of its neighboring nodes [54]. Often sensor *data attributes* are correlated, e.g., temperature has certain correlation with humidity. On the other hand, sensor data collected in densely deployed WSNs tends to be correlated in both time and space. This is very true in case of environmental monitoring applications [28]. *Spatial correlations* mean that sensor observations collected from geographically close sensor nodes are highly similar, while *temporal correlations* indicate consecutive sensor observations collected from a sensor node are highly similar [57].

2.2.2 Application-Dependent Issues

Applications pose different requirements on outlier detection techniques, as different applications may have different definitions and characteristics for outliers. Here we address application-dependent issues that need to be taken into account while designing an outlier detection technique for WSNs.

- **Local outlier vs. Global outlier**. *Local outliers* represent those outliers that are detected at individual sensor node only using its local data. *Global outliers* represent those outliers that are detected in a more global perspective [104] by considering a cluster of sensor nodes. Specifically, global outliers can be identified at a parent node, cluster-head node, or even a central station, by collecting many data from its assigned sensor nodes. Alternatively, global outliers can be identified at individual sensor node using a well-defined normal behavior of sensor data, which is modelled in a global view. One should note that a local outlier may not be identified as a global outlier and vice versa [118]. The choice between identifying local outliers and global outliers depends on the requirements of applications.
- **Error vs. Event**. Semantic of outliers depends on application at hand. Errors are those sensor observations that do not conform to the true state

of monitoring phenomena and significantly deviate from the priori defined normal behavior of sensor data. Events, on the other hand, are those sensor observations that do not conform to the priori defined normal behavior of sensor data but reflect the true state of monitoring phenomena. In fact, distinguishing between errors and events is not very simple, except for *absolute errors*, which have extremely high or low values, and these extreme values are usually impossible to occur in real-life. Errors also may be *random errors* or *long-term errors*. Random errors usually randomly occur at a very short time period while long-term errors last for a relatively long period of time. These two types of errors may not show extreme values like absolute errors but do deviate from the defined normal behavior of sensor data. Moreover, long-term errors may represent similar values with events so that it is hard to distinct between errors and events only by analyzing sensor data of a node itself [141].

- **Degree of being an outlier**. A sensor observation can be simply labelled as an outlier manually or by setting a threshold. However, a more thorough outlier detection technique extensively analyzes sensor data. A straightforward way for this analysis is to define a normal behavior for sensor data and consider those sensor observations that deviate from the defined normal behavior of sensor data as outliers [119]. The normal behavior of sensor data can usually be modelled by a *normal boundary* [18]. The normal boundary can be modelled using different methods, e.g., *confidence level* [10] or *well-defined shape* [118]. One should note that the defined boundary representing the normal behavior of sensor data may evolve over time.

- **Handling outlier**. We have categorized outliers into errors and events. Strategies on handling these outliers depends on applications. Errors, especially absolute errors, significantly influence sensor data quality and thus need to be instantly removed or corrected using predicted values [10]. Due to the fact that events contain important information about state of the phenomena and also change the priori defined normal behavior of sensor data, they need to be used to model new normal behavior and generate a notification specifying occurrence of the event.

- **Distributed vs. Centralized processing**. Distinction between *distributed* and *centralized* outlier detection techniques refer to where and how outlier detection is performed. Distributed outlier detection techniques identify outliers at individual sensor node, while centralized outlier detection techniques identify outliers at a parent node, or a cluster-head node, or even a central station. Compared to centralized manner, distributed man-

ner of identifying outliers locally on sensor nodes reduces the transmission of raw sensor observations and makes efficient use of network resources such as energy and bandwidth. However, the accuracy of outlier detection techniques using distributed manner may not be as good as centralized manner due to lack of enough sensor data for the modelling purpose.

- **Online detection vs. Offline detection**. Online outlier detection techniques identify outliers (near) real-time whenever and wherever they occur. Offline outlier detection techniques identify outliers only when large volume of observations are collected for a relatively long period of time. Compared to offline manner, online manner of identifying outliers in real-time reduces the detection delay and can quickly detect occurred events (suitable for real-time WSN applications). However, false alarm rate of outlier detection techniques using online manner may be higher than offline manner due to lack of enough temporal information representing nature and type of outlier.

2.2.3 Performance Metrics

After considering sensor data characteristics and WSN application-dependent issues, we provide two important performance metrics, i.e., *detection accuracy* and WSNs *resource consumption* to evaluate outlier detection techniques. The detection accuracy itself is composed of *detection rate* and *false alarm rate*. WSNs resource consumption relates to communication, computational, and memory complexity. Of course, there is a trade-off between these two performance metrics.

- **Detection rate & False alarm rate**. The detection rate is the percentage of outliers that are correctly identified and is represented by the ratio between number of correctly identified outliers and total number of outliers. The false alarm rate is the percentage of normal observations that are incorrectly considered as outliers and is represented by the ratio between the number of normal observations that are incorrectly considered as outliers and total number of normal observations. An effective outlier detection technique is required to maintain a high detection rate while keeping the false alarm rate low. The trade-off between detection rate and false alarm rate can be represented by *receiver operating characteristic* (ROC) curves [65]. Figure 2.1 illustrates ROC curves for different outlier detection techniques, and the performance of outlier detection techniques can be represented by the area under the ROC curve (AUC). The larger the AUC, the better the performance of the outlier detection technique.

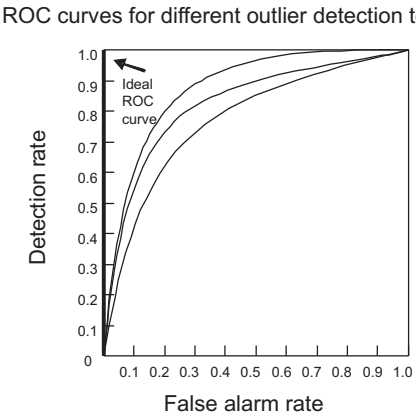


Figure 2.1: ROC curves for different outlier detection techniques [65]

- **Communication, Computational, & Memory Complexity**. An efficient outlier detection technique for WSNs should have low communication, computational and memory complexity (usually represented by $O()$).

2.3 Shortcomings of General Outlier Detection Techniques

We have presented several important issues to be taken into consideration while designing outlier detection techniques for WSNs. Now, we briefly review general outlier detection techniques categorized based on techniques they use as well as their degree of using pre-lablled data [118]. We then highlight their shortcomings and explain why they are not directly applicable for WSNs.

General outlier detection techniques are usually designed for *simple dataset*, which assumes data has no complex semantics and can be represented by low-dimensional real-value attributes [10]. Moreover, general outlier detection techniques are not designed for specific application domains. Based on types of technique they use, general outlier detection techniques can be categorized as:

- **Distribution-Based**. *Distribution-based techniques* [35, 43, 10, 96, 29] assume that the entire dataset conforms to a standard statistical distribution model and determine a data point as an outlier depending on whether the point deviates significantly from the data model. These techniques can fast

and effectively identify outliers on the basis of an appropriate probabilistic data model. However, they are not suitable to identify outliers in even moderately high dimensional spaces and suffer from the fact that a priori knowledge of data distribution is not available in many real-life situations.

- **Depth-Based**. *Depth-based techniques* [117, 96, 95, 58] use the concept of computational geometry and organize data points in layers in multi-dimensional data spaces, in which each data point is assigned a depth. Outliers are considered to be those points in the shallow layers with smaller depth values. These techniques avoid the problem of fitting the entire dataset into a single data distribution, but are inefficient for large datasets with high dimensionality.

- **Graph-Based**. *Graph-based techniques* [66, 89, 124] map the dataset into a graph to visualize the single or multi-dimensional data spaces, e.g., *box plot* or *scatter plot* [118]. Outliers in these techniques are those points that are present in particular positions of the graph. These techniques have no assumption about the data distribution and instead exploit the graphical representation of the data to visually highlight outliers. However, they are limited due to lack of precise criteria to detect outliers.

- **Clustering-Based**. Traditional *clustering-based techniques* [30, 140, 37, 32] are developed to optimize the process of clustering of data and outlier detection is only a by-product of no interest. The novel clustering-based outlier detection techniques [136, 56, 49, 94] can effectively identify outliers as points that do not belong to clusters or as clusters that are significantly smaller than other clusters. However, these techniques are suspectable to high dimensional datasets since they rely on the full-dimensional distance measure of points in clusters.

- **Distance-Based**. *Distance-based techniques* [61, 97, 9, 5] are used to identify outliers based on the measure of full dimensional distance between a point and its nearest neighbors in a dataset. Outliers in these techniques are those points that are distant from the neighboring points in the dataset. These techniques do not make any assumptions about the data distribution and have better computational efficiency than depth-based techniques, especially in large datasets. However, they rely on the existence of some well-defined notions of distance and do not work well in high dimensional datasets. Also, they cannot discover local outliers, especially in datasets with diverse densities and arbitrary shapes.

- **Density-Based**. *Density-based techniques* [8, 16, 59, 46, 87, 63, 33, 62] are proposed to take the local density into account when searching for outliers. The computation of density still depends on full dimensional distance measure between a point and its nearest neighbors in a dataset. These techniques can effectively identify local outliers in datasets with diverse clusters. However, they need to determine appropriate input parameters and do not work well in high dimensional datasets.

- **Neural network-based**. *Neural networks-based techniques* [105, 50, 34] can autonomously model the underlying data distribution and distinguish between normal and abnormal classes. In these techniques, data points that are not reproduced well at the output layer are considered as outliers. These techniques effectively identify outliers and automatically reduce the input features based on the key attributes. However, they are still susceptible to high dimensional datasets and also sensitive to model parameters.

- **Bayesian network-based**. *Bayesian network-based techniques* [5, 12] can estimate the posterior probability of observing a class for each test data and identify data points as outliers if the likelihood of these points belonging to their expected class is rather low. These techniques effectively identify outliers by incorporating both a prior knowledge and data points. However, they are sensitive to pre-defined behavior of data points and also have considerably high computational cost.

- **Support vector machine-based**. *SVM-based techniques* [106, 121, 122, 90] can distinguish between normal and abnormal classes by mapping data into the feature space. Those points that are distant from most other points or are in relatively sparse regions of the feature space are declared as outliers in these techniques. These techniques efficiently identify outliers by using the kernel functions. However, they have considerably high computational cost for the computation of kernel functions. Moreover, it is not easy to determine an appropriate parameter to control the size of boundary region.

Another categorization of general outlier detection techniques can be done based on the degree of using pre-labelled data [118]. This categorization results in *supervised*, *unsupervised*, and *semi-supervised* categories:

- **Supervised**. These techniques require pre-labelled data to learn a normal and an abnormal model and then classify a new data point as normal or outlier depending on which model the data point fits into. These techniques build classifiers to distinguish between normal and known outliers and are popular in fraud detection and intrusion detection applications. However,

pre-labelled data is not easy to obtain in many real-life applications and also new types of rare events may not be included in pre-labelled data.

- **Unsupervised**. These techniques require no pre-labelled data and use certain criteria to identify outliers, e.g., similarity measure between a point and its nearest neighboring points. Compared to supervised techniques, unsupervised techniques are more general as they do not need a prior knowledge about data and its distribution and can work on unlabelled data.

- **Semi-Supervised**. These techniques require no pre-labelled abnormal data but require pre-labelled normal data to learn normal behavior and then to classify a new data point as normal or outlier depending on how well the data points fit into the normal model. Many semi-supervised techniques can be adapted to operate in an unsupervised mode using a sample of the unlabelled dataset as training data. Such adaptation assumes that the test data contains very few anomalies and the model learnt during training is robust to these few anomalies.

The general outlier detection techniques are not directly applicable to outlier detection in WSNs due to their following major *shortcomings*:

- They are usually designed for *static data*, i.e., all data points are collected and no new data point will be inserted. This is contradictory with nature of distributed streaming sensor data.

- They usually ignore the correlations of data, i.e., attribute correlation, spatial correlation and temporal correlation. This makes them unsuitable to be applied on correlated sensor data.

- The pre-lablled data required in supervised and semi-supervised techniques is not always available in WSNs.

- They usually do not distinguish between errors and events and only regard detected outliers as errors. Also, they usually have no specific strategy for handling identified outliers, only find them or remove them all. This leads to loss of important information about events occurred in the network and make these techniques not suitable for event-driven WSN applications.

- They usually identify outliers in a centralized and offline manner. This is not suitable for real-time WSN applications due to high communication overhead and long detection delay.

- They have paid little attention to computational and memory complexity. In general, they are computationally expensive and require large memory capacity. This makes them not suitable for resource-constraint WSNs.

2.4 Taxonomy of Outlier Detection Techniques for WSNs

We have highlighted the shortcomings of general outlier detection techniques that make them not suitable for WSNs and sensor data. In recent years, outlier detection has also attracted more and more attention in WSNs field, and many outlier detection techniques specifically developed for WSNs have emerged [146]. In this section, we provide a technique-based taxonomy framework to categorize existing outlier detection techniques designed for WSNs. We classify them based on the disciplines from which they adopt their ideas and address the key characteristics and performance analysis of each outlier detection technique using this taxonomy. Furthermore, we provide a brief evaluation for each of these disciplines.

As illustrated in Figure 2.2, we categorize outlier detection techniques developed for WSNs into *statistical-based*, *nearest neighbor-based*, *clustering-based*, *classification-based*, and *spectral decomposition-based* techniques. Statistical-based techniques are further categorized into *parametric-based* and *non-parametric based* techniques based on how the probability distribution model is built [71]. *Gaussian model-based* and *non-Gaussian model-based* techniques belong to parametric-based techniques, and *kernel function-based* and *histogram-based* techniques belong to non-parametric based techniques. Classification-based techniques are categorized as *Bayesian network-based* and *support vector machine-based* techniques based on type of classification model that they use. Bayesian network-based techniques are further categorized into *naive Bayesian network-based*, *Bayesian belief network-based*, and *dynamic Bayesian network-based* techniques based on the degree of probabilistic independencies among variables. Spectral decomposition-based techniques use *principle component analysis* for outlier detection.

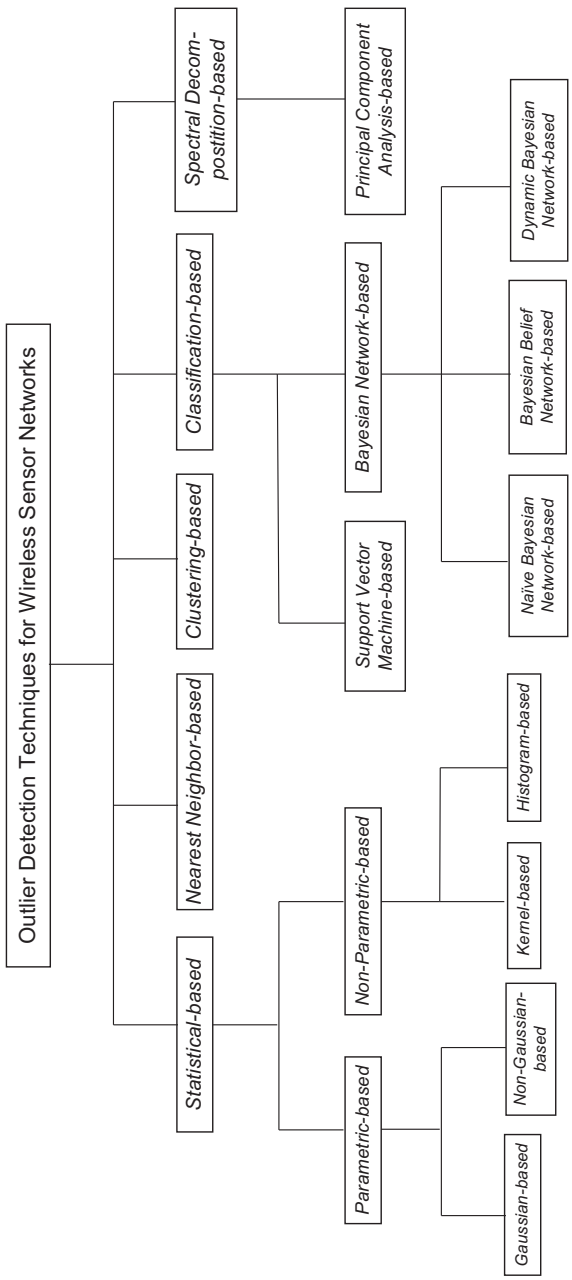


Figure 2.2: Taxonomy of outlier detection techniques for WSNs

2.4.1 Statistical-Based

Statistical-based techniques are the earliest techniques to deal with the problem of outlier detection. Statistical outlier detection techniques are essentially *model-based* techniques. They assume that normal data instances occur in high probability regions of a statistical distribution, while outliers occur in the low probability regions of the statistical distribution. They first fit a statistical (probability distribution) model to the given data and then apply a statistical inference test to determine whether a new data instance belongs to this model or not. A data instance is declared as an outlier if it has a low probability. Statistical-based techniques can work in an unsupervised mode, in which a statistical model can be defined when small number of data points are outliers and the majority of the observations can fit into the model [18]. Statistical-based techniques are categorized into parametric and non-parametric based on how to fit the statistical model.

Parametric-Based

Parametric-based techniques assume availability of knowledge of underlying data distribution, i.e., the data is generated from a known parametric distribution. They then estimate the distribution parameters from the given data. Based on type of distribution assumed, these techniques are further categorized into Gaussian model-based and non-Gaussian model-based techniques. In Gaussian model-based techniques, the data is assumed to be generated from a Gaussian distribution.

- **Gaussian model-based**. Wu et al. [131] present two local techniques for identification of outlying sensors as well as identification of event boundary in sensor networks. These techniques employ the spatial correlation of the readings existing among neighboring sensor nodes to distinguish between outlying sensors and event boundary. In the technique for identifying outlying sensors, each node computes the difference between its own reading and the median reading from its neighboring readings. Then it standardizes all differences from its neighborhood. A node is considered as an outlying node if the absolute value of its reading's deviation degree is sufficiently larger than a pre-selected threshold. The technique of event boundary detection is based on the previous results of outlying sensor identification and determines a node as an event node if the absolute value of the node's deviation degree in one geographical region is much larger than that in another region. Accuracy of these outlier detection techniques is not relatively high due to the fact that they ignore the temporal correlation of sensor readings.

Bettencourt et al. [7] present a local outlier detection technique to identify errors and detect events in ecological applications of WSNs. This technique can distinguish between erroneous measurements and events using the spatio-temporal correlations of sensor data. Each node learns the statistical distribution of difference between its own measurements and each of its neighboring nodes, as well as between its current and previous measurements. The procedure can be based on a priori knowledge of data distribution or a non-parametric density estimation. A measurement is identified as anomalous if it is less than a user-specified threshold in the statistical significance test. The detected anomalous measurement may be considered as event if it is likely to be temporally different from its previous measurements, but it is spatially correlated. The drawback of this technique is that it relies on the choice of the appropriate values of the threshold.

Hida et al. [48] design a local technique to make simple aggregation operations, such as MAX or AVG, more reliable under presence of faulty sensor readings and failed nodes. This technique relies on the spatio-temporal correlations of sensor data and uses two statistical tests to locally detect outliers. Each incoming sensor value is compared against the current value and the previous values of all nodes in the neighborhood. If the incoming value passes the two statistical tests, it is allowed to be aggregated as usual; otherwise (if the incoming value is outside of 2.5 standard deviations of the mean) it is declared as an outlier and will be eliminated from the analysis. Drawbacks of this technique include the fact that it only deals with one-dimensional outlier data and nodes require large memory to store historical values of all its neighboring nodes.

- **Non-Gaussian model-based**. Jun et al. [55] present a statistical-based technique, which uses a symmetric α-stable (SαS) distribution to model outliers being in form of impulsive noise. The technique utilizes the spatio-temporal correlations of sensor data to locally detect outliers. Each node in a cluster first detects and corrects temporal outliers by comparing the predicted data and the sensed data. Then the cluster-head collects the rectified data from all other nodes in the cluster and further detects spatial outliers that deviate remarkably from other normal data. This technique reduces computational cost as the cluster-heads carry out most of the computation tasks. However, this technique identifies spatial outliers only when all actual data of nodes is collected for a long period of time. This centralized and offline manner has high communication cost and long detection delay.

Non-Parametric Based

Non-parametric based techniques do not assume availability of data distribution or any a priori knowledge about data parameters. They instead estimate the density of the distribution from the given data. Two most widely used techniques in this category are histogram-based and kernel function-based. Histogram-based techniques are also referred to as *frequency-based* or *counting-based* techniques. They first count frequency of occurrence of different data instances (thereby estimating the probability of occurrence of a data instance) and compare a test instance with each of the categories in the histogram and test whether it belongs to one of them. If the test instance does not belong to any category, it is considered as outlier. Kernel function-based techniques use kernel functions to estimate the probability distribution function (PDF) for the normal instances. A new instance that lays in the low probability area of this PDF is declared as an outlier.

- **Histogram-Based**. Sheng et al. [102] present a histogram-based technique to identify global outliers in data collection applications of sensor networks. This technique attempts to reduce communication cost by collecting histogram information rather than collecting raw data for centralized processing. The sink uses histogram information to extract data distribution from the network and to filter out the non-outliers. Outliers can be identified by re-collecting more histogram information from the network. The identification of outliers is achieved by a fixed threshold distance or the rank among all outliers. A drawback of this technique is that re-collecting more histogram information from the whole network will cause high communication overhead and long detection delay.

- **Kernel function-based**. Palpanas et al. [88] propose a kernel-based technique for online identification of outliers in streaming sensor data. This technique requires no a priori data distribution and uses kernel density estimator to approximate the underlying distribution of sensor data. Thus, each node can locally identify outliers if the values deviate significantly from the model of approximated data distribution. An observation is considered as an outlier if the number of values being in its neighborhood is less than a user-specified threshold. This technique can be extended to high-level nodes for identification of outlier in a more global manner. The main problem of this technique is its high dependency on the defined threshold, as choice of an appropriate threshold is quite difficult and a single threshold may also not be suitable for outlier detection in multi-dimensional data. Furthermore, the technique does not consider maintaining the model while

sensor data is frequently updated.

Subramaniam et al. [104] further extend the work of Palpanas et al. [88] and solve the two previous problems of insufficiency of a single threshold for multi-dimensional data and maintaining the data model built by kernel density estimator. They propose two global outlier detection techniques for complex applications. One technique allows each node to locally identify outliers using the same technique as Palpanas et al. [88] and then transmit the outliers to its corresponding parent to be checked until the sink eventually determines all global outliers. In the other technique, each node employs a more robust technique called LOCI [87] to locally detect global outliers by having a copy of global estimator model obtained from the sink. Experimental results show that these techniques achieve high accuracy in terms of estimating data distribution and high detection rate while having low memory usage and message transmission overhead.

Evaluation of Statistical-Based Techniques

Statistical-based techniques are mathematically justified and can effectively identify outliers if a correct probability distribution model is acquired. Moreover, after constructing the model, the actual data on which the model is based on is not required. However, a priori knowledge required by parametric techniques is often not available or is expensive to compute in many real-life WSNs applications. Thus parametric techniques may not be useful if sensor data does not follow the assumed distribution. Non-parametric techniques are more flexible and autonomous due to the fact that they do not make any assumption about the distribution characteristics. The computational complexity of statistical-based techniques depends on the nature of statistical model that is required to be fitted on the data. Histogram-based techniques are very efficient for univariate data and are relatively simple to implement, but they are not able to capture the interactions between different attributes of multivariate data. Also, it is not easy to determine an optimal size of the bins to construct the histogram. Kernel function-based techniques can scale well in multivariate data but potentially have quadratic time complexity in terms of the data size.

2.4.2 Nearest Neighbor-Based

Nearest neighbor-based techniques are popular within the data mining community to use several well-defined distance notions to compute the distances (*similarity measure*) between two data instances [61, 97]. The key assumption of these techniques is that normal data instance has close neighbors while outliers are

located far from other data instances. They first compute neighborhood for each data instance and then analyze the neighborhood to determine whether a data instance is an outlier, i.e., being located far from its neighbors.

Branch et al. [6] propose a technique based on distance similarity to identify global outliers in sensor networks. This technique attempts to reduce the communication overhead by a set of representative data exchanges among neighboring nodes. Each node uses distance similarity to locally identify outliers and then broadcasts the identified outliers to neighboring nodes for verification. The neighboring nodes repeat the procedure until all of the sensor nodes in the entire network eventually agree on the global outliers. Since, the technique does not adopt any network structure, i.e., every node uses broadcast to communicate with other nodes in the network, it will have high communication overhead. Consequently, it does not scale well to the large-scale networks.

Zhang et al. [138] propose a distance-based technique to identify n global outliers in snapshot and continuous query processing applications of sensor networks. This technique reduces communication overhead as it adopts the structure of aggregation tree and prevents broadcasting of each node in the network [6]. Each node in the tree transmits some useful data to its parent after collecting all the data sent from its children. The sink then roughly figures out top n global outliers and sends these outliers to all the nodes in the network for verification. If any node disagrees on the global results, it will send extra data to the sink again for outlier detection. This procedure is repeated until all the nodes in the network agree on the global results calculated by the sink. This technique causes too long detection delay and high communication overhead.

Zhuang et al. [141] present two in-network outlier cleaning techniques for data collection applications of sensor networks. One technique uses wavelet analysis specifically for outliers such as noises or occasionally appeared errors. The other technique uses dynamic time warping (DTW) distance-based similarity comparison specifically for outliers that are erroneous and last for a certain period of time. In this technique, each node transforms raw data into the wavelet time-frequency domain and identifies the high-frequency data measurements as outliers and corrects them using proper wavelet coefficients. The long segmental outliers can be detected and removed by comparing the similarity of two sensing series of the neighboring nodes within two forwarding hops. A drawback of this technique, however, is its dependency on a suitable pre-defined threshold that is not easy to define.

Evaluation of Nearest Neighbor-Based Techniques

Nearest neighbor-based techniques are unsupervised in nature and do not make any assumption regarding data distribution. They can also generalize many notions from statistical-based techniques. However, these techniques suffer from the choice of the appropriate input parameters. Moreover, defining distance measures between instances is challenging for sensor data. Additionally, in multivariate data sets it is computationally expensive to compute the distance between data instances and as a result these techniques lack scalability.

2.4.3 Clustering-Based

Clustering-based techniques are popular within the data mining community to group similar data instances into *clusters* with similar behavior. They assume that normal data instances belong to large and dense clusters, while outliers do not belong to any significant cluster. They first cluster data into a finite number of clusters and then analyze each data instance with respect to its closest cluster. Those data instances that do not belong to clusters and those clusters that are significantly smaller than other clusters are considered as outliers.

Rajasegarar et al. [98] propose a global outlier detection technique based on clustering technique to identify anomalous measurements in sensor nodes. This technique minimizes the communication overhead by clustering the sensor measurements and merging clusters before communicating with other nodes. Initially, each node clusters the measurements and reports cluster summaries rather than transmitting the raw sensor measurements to its parent. The parent then merges cluster summaries collected from all of its children before sending them to the sink. An anomalous cluster can be determined in the sink if the cluster's average inter-cluster distance is larger than one threshold value of the set of inter-cluster distances. Determining the parameter k (the k nearest neighbor clusters), which is used to compute the average inter-cluster distance is not always easy. The parameter of cluster width may also not be defined appropriately. Furthermore, the identification of anomalous clusters may trigger a false alarm.

Evaluation of Clustering-Based Techniques

Clustering-based techniques are unsupervised in nature and do not require a priori knowledge about data distribution. They are capable of being used in an incremental model, i.e., new data instance can be incorporated into the system and being tested to find outliers. However, the performance of these techniques is highly dependent on the effectiveness of capturing the cluster structure of normal

instances. Additionally, similar with nearest neighbor-based techniques, computing the distance between multivariate data instances using clustering-based techniques is computationally expensive.

2.4.4 Classification-Based

Classification-based techniques are popular in the data mining and machine learning communities. They learn a classification model (*classifier*) using a set of data instances (*training*) and classify an unseen instance into one of the learned (normal/outlier) classes (*testing*). The unsupervised classification-based techniques require no knowledge about available labelled training data and learn the classification model, which fits the majority of the data instance during training. The one-class unsupervised techniques learn the boundary around the normal instances while some anomalous instance may exist and declare any new instance falling outside this boundary as an outlier. The classifier may need to update itself to accommodate new instances that belong to the normal class. Existing classification-based outlier detection techniques for WSNs are categorized into support vector machine (SVM)-based and Bayesian network-based techniques based on type of classification model they use.

Support Vector Machine-Based

SVM-based techniques separate the data belonging to different classes by fitting a hyperplane between them which maximizes the separation. The data is mapped into a higher dimensional feature space by using a non-linear function where it can be easily separated by a *hyperplane*. Furthermore, a *kernel function* is used to approximate the inner product between the mapped vectors in the feature space to find the hyperplane. One-class SVM-based techniques do not require any labels for normal or outlier data and allow existence of outliers in the training set. They learn a normal boundary which encloses the majority of data vectors and detect new unseen data as outliers if they deviate from the normal boundary.

Rajasegarar et al. [99] propose a SVM-based technique for detecting outliers in sensor data. This technique uses quarter-sphere one-class SVM to reduce computational complexity of quadratic optimization and to locally identify outliers at each node. The sensor data that lays outside the quarter sphere is considered as an outlier. Each node communicates only summary information (the radius information of sphere) with its parent for global outlier classification. This technique identifies outliers only when all data measurements are collected for a long period of time and therefore causes detection delay. Moreover, modelling the quarter-sphere SVM in the feature space has a high memory complexity.

Bayesian Network-Based

Bayesian network-based techniques use a *probabilistic graphical model* to represent a set of variables and their probabilistic independencies. They aggregate information from different variables and provide an estimate on the expectancy of a data instance to belong to the learned class. They are categorized as naive Bayesian network-based, Bayesian belief network-based, and dynamic Bayesian network-based techniques based on degree of probabilistic independencies among variables. Naive Bayesian network-based techniques capture spatio-temporal correlations among sensor nodes. Bayesian belief network-based techniques consider the correlations among the attributes of the sensor data. Dynamic Bayesian network-based techniques consider the dynamic network topology that evolves over time by adding new state variables to represent the system state at the current time instance.

- **Naive Bayesian network-based**. Elnahrawy and Nath [28] present a Bayesian model-based technique to discover outliers and detect outlying sensors. This technique maps the problem of learning spatio-temporal correlations to the problem of learning the parameters of the Bayesian classifier and then uses the classifier for probabilistic inference. Each node locally computes the probabilities of each of its incoming readings being in all subintervals (classes) divided from the whole values interval. If the probability of a reading in its class is smaller than that of being in other classes, it is considered as an outlier. The technique requires no user-specified threshold to determine outliers and can also be used to approximate the missing readings occurred in the network. It, however, does not specify how to decide a specific spatial neighborhood and only assumes that there are two neighbors surrounding. Also, this technique does not use the spatial and temporal correlations for outlier detection.

- **Bayesian belief network-based**. Janakiram et al. [54] present a technique based on Bayesian belief network (BBN) to identify outliers in streaming sensor data. This technique uses BBN to capture not only the spatio-temporal correlations that exist among the observations of sensor nodes, but also conditional dependency among the observations of sensor attributes. Each node trains a BBN to detect outliers based on behaviors of its neighbors' readings as well as its own reading. An observation is considered as an outlier if it falls beyond the range of the expected class. Compared to naive Bayesian networks, this technique improves the accuracy in detecting outliers as it considers conditional dependencies among the attributes. Accuracy of a BBN depends on degree of conditional dependency among

the observations of sensor attributes. This technique may not work well in presence of the dynamic network topology change.

- **Dynamic Bayesian network-based**. Hill et al. [44] present two techniques based on dynamic Bayesian networks (DBNs) to identity outliers in environmental sensor data streams. This technique uses DBNs to fast track changes in dynamic network topology of sensor networks. One technique assumes that there is only a measured state variable existing in the multivariate data and the current state can be determined only depending on its historical state. This technique identifies outliers by computing the posterior probability of the most recent data values in a sliding window. The data measurements that fall outside the expected value interval are considered as outliers. The other technique uses a more complex DBN including two measured state variables for outlier detection. This technique makes it possible to operate on several data streams at once.

Evaluation of Classification-Based Techniques

Classification-based techniques provide an exact set of outliers by building a classification model to distinguish between instances belonging to different classes. Moreover, the testing phase is fast since each test instance only needs to be compared with the pre-computed model. However, a main drawback of SVM-based techniques is their computational complexity of quadratic optimization and the choice of proper kernel functions. Learning the accurate classification model of a Bayesian network is challenging for Bayesian network-based techniques if number of variables is large.

2.4.5 Spectral Decomposition-Based

Spectral decomposition-based techniques try to find combination of attributes that capture the behavior of sensor data well. They assume that data can be embedded into a low dimensional subspace, in which normal instances and outliers appear to be significantly different. Principal component analysis (PCA) is commonly used to reduce dimensionality before outlier detection and to finds new subset of dimension which capture the behavior of the data. Specifically, the top few principal components capture degree of variability and any data instance that violates this structure for the smallest components is considered as an outlier.

Chatzigiannakis et al. [19] propose a PCA-based technique to solve data integrity and accuracy problem caused by compromised or malfunctioning sensor nodes. This technique uses PCAs to efficiently model the spatio-temporal data correlations in a distributed manner and identifies outliers spanning through

neighboring nodes. Each primary node builds a model of the normal condition offline by selecting appropriate principal components (PCs) and then obtains sensor readings from other nodes in its group and performs local real-time analysis. Sensor measurements that significantly vary from the modelled variation value under normal condition are declared as outliers. The primary nodes eventually forward the information about outlier data to the sink. The offline procedure of selecting appropriate PCs is computationally very expensive.

Evaluation of Spectral Decomposition-Based Techniques

Since PCA-based techniques tend to capture the normal pattern of the data using a subset of dimensions, they can be applied to high-dimensional data. They can operate in an unsupervised mode. However, selecting suitable PCs, which is needed to accurately estimate the correlation matrix of normal patterns, is computationally very expensive. Furthermore, they are useful only if outliers and normal instances are highly distinguishable in the reduced space.

2.5 Guideline of Outlier Detection Techniques for WSNs

In this section, we provide a comparative table to compare the above described outlier detection techniques specifically developed for WSNs in terms of important considerations for outlier detection techniques for WSNs addressed in Section 2.2. We further present a guideline on requirements of outlier detection techniques for WSNs.

Techniques	Sensor Data Characteristics					Application-Dependent Issues				Complexity	
	Attribute		Correlation			Local/ Global Outlier	Error/ Event	Handling Outlier	Distributed/ Centralized	Online/ Offline	Communication	Computation and Memory
	Univariate	Multivariate	Attribute	Spatial	Temporal							
Wu et al. [131]	√			√		Global	√		Distributed	Online	Medium	Medium
Bettencourt et al. [7]	√			√	√	Global	√		Distributed	Online	Medium	Medium
Hida et al. [48]	√			√	√	Global			Distributed	Online	Medium	Medium
Jun et al. [55]	√			√	√	Global			Centralized	Offline	Medium	Medium
Sheng et al. [102]	√					Global			Centralized	Offline	High	Low
Palpanas et al. [88]	√				√	Local			Local	Online	Low	medium
Subramaniam et al. [104]		√			√	Global			Distributed	Online	Medium	Medium
Branch et al. [6]	√					Global			Centralized	Offline	High	Medium
Zhang et al. [138]	√					Global			Centralized	Offline	High	Medium
Zhuang et al. [141]	√			√		Local			Distributed	Offline	Medium	Medium
Rajasegarar et al. [98]		√				Global			Centralized	Offline	Medium	Medium
Rajasegarar et al. [99]		√				Global			Distributed	Offline	Low	Medium
Elnahrawy and Nath [28]	√			√	√	Global			Distributed	Online	Medium	Medium
Janakiram et al. [54]		√	√	√	√	Global			Distributed	Online	Medium	High
Hill et al. [44]		√		√	√	Global			Distributed	Online	Medium	High
Chatzigiannakis et al. [19]		√		√	√	Global			Distributed	Offline	Medium	High

Table 2.1: Comparison of outlier detection techniques for WSNs

Table 2.1 presents a comparison between outlier detection techniques developed specially for WSNs, from which we can summarize the characteristics of current outlier detection techniques for WSNs as:

- They mostly identify global outliers.
- They more recently take spatial and temporal correlations of sensor data into account.
- They mostly identify outliers in a distributed manner, while less attention has been paid to online outlier detection.
- They have average communication, computational, and memory complexity.
- They mostly consider univariate data, rather than multivariate data, or correlation of data attributes.
- They hardly consider how to distinguish between errors and events and have no strategy to handle different types of outliers during outlier detection.

By identifying these shortcomings and in line with important issues to be considered while designing outlier detection techniques for WSNs, we present a *guideline* on requirements that a suitable outlier detection technique for WSNs should meet:

- It should identify outliers in a distributed manner while having a low data transmission.
- It should identify outliers in real-time (i.e., having low detection delay).
- It should update the modelled normal behavior of sensor data over time to cope with dynamic nature of sensor data.
- It should take multivariate data into account and consider correlation among attributes of sensor data.
- It should be unsupervised not to need labelled data.
- It should consider spatial and temporal correlations among sensor data.
- It should scale well with the increase of network size.
- It should effectively distinguish between erroneous measurements and events and appropriately handle them in real-time.

- It should maintain a good balance between detection rate and false alarm rate.
- It should be cheap and have low computational and memory complexity.

2.6 Chapter Summary

In this chapter, we address several important issues to be considered while designing outlier detection techniques for WSNs and explain why general outlier detection techniques are not directly applicable for WSNs. We provide a technique-based taxonomy of those outlier detection techniques specifically developed for WSNs and compare them in a comparative table. We further present a guideline of requirements of an optimal outlier detection technique for WSNs.

According to the analysis of general outlier detection techniques, and the overview of current outlier detection techniques for WSNs, we conclude that there is no generic outlier detection technique applicable for all application domains or data types. Also, no existing outlier detection technique considers all issues of outlier detection and satisfies all requirements. As we will show in the coming chapters, we take into account all the requirements presented in the guideline for an optimal outlier detection technique for WSNs while designing our techniques. The performance evaluation and design choices we make show that these techniques fulfill the specified requirements.

Chapter 3

Sensor Data Labelling Techniques

To measure the performance of an outlier detection technique, one needs a reference value, usually called a ground truth. Often, labelling techniques are used to label sensor data and classify each data point as normal or outlier. The choice of the labelling technique strongly influences the performance of outlier detection techniques. Therefore, it is important to choose the right technique for labelling a dataset before performing outlier detection. In doing so, the shape of the dataset as well as the definition that application at hand uses for outliers are two deciding factors on what labelling techniques should be used. In this chapter, we first define different types of outliers and then investigate performance of four different labelling techniques based on, i.e., Mahalanobis distance, density, running average, and Bayesian networks, to identify them. To present impact of labelling techniques on outlier detection process, we will use the dataset labelled using these techniques in the following chapters.

3.1 Introduction

As it has been shown in the previous chapters, the term *outlier* can be defined in many different ways, depending on the context and the outlier detection technique used. The definition of an outlier depends on the application and on the characteristics of sensor data to be analyzed.

To evaluate the performance of an outlier detection technique, one needs a reference value, usually called a *ground truth*. However, quite often the ground truth is not available. Therefore, labelling techniques are used to label sensor data and assign each data point to a *normal* or *outlier* class. Due to the various possible interpretations of the term outlier and the fact that one labelling technique might work well for one dataset, while it performs badly on another, it is hard to choose a suitable labelling technique. The characteristics of the dataset as well as the labelling technique are the two deciding factors in this selection. To complicate the matter, one should note that an outlier detection technique might have a very high detection rate on results of one labelling technique, while fails when used for another.

To clarify, let us assume we use a clustering-based technique to label a dataset and then use a time series streaming based outlier detection technique to identify outliers. Clustering-based labelling techniques consider outliers to be either data points that do not belong to clusters or clusters that are significantly smaller than other clusters [136, 56]. Time series streaming based outlier detection techniques, however, state that if the removal of a point from the time sequence results in a sequence that can be represented more briefly than the original one, then the point is an outlier [75]. It is obvious that these two definitions of outliers have very little in common, which results in failing the outlier detection technique to correctly identify outliers labelled by the used labelling technique. On the one hand, a dataset is usually not labelled solely for the purpose of the outlier detection and many other applications will use it. On the other hand, often no information about what labelling techniques have been used to label data is available. Therefore, it is necessary to have a guideline on circumstances under which each labelling technique can identify outliers.

Since there exists no universally accepted definition for an outlier, there is also no general purpose labelling technique. In this chapter we investigate and compare four data labelling techniques based on *Mahalanobis distance*, *density*, *running average*, and *Bayesian networks*. This results in identification of various types of outliers occurring in sensor dataset, which will be identified in Section 3.2. The real dataset used by our labelling techniques are described in Section 3.3. Detailed explanation of the four different labelling techniques used in this chapter is provided in Section 3.4. We present a thorough comparison between these

labelling techniques in terms of performance, complexity, and the effect of the data characteristics in Section 3.5. Based on these comparisons we present a guideline on choosing the labelling technique which best fits the characteristics of the outlier detection techniques presented in later chapters in Section 3.6. Finally, this chapter is concluded in Section 3.7.

3.2 Types of Outliers

As presented before the semantic of an outlier greatly varies and covers a wide range of different events and errors. Within this semantic, we identify the following four types of outliers, as illustrated in Figure 3.1:

- Type 1: *Incidental absolute errors.* Having isolated (one-time spike) or very short sequence of extreme high or low values are often indication of absolute errors. An example of such values is a temperature reading of 100 °C in a mountainous location at a certain time instance. This type of outliers can easily be identified using a pre-defined threshold. For instance, in case of the temperature readings in the mountains in Switzerland, reasonable minimum and maximum values can be found using mythological data of the last ten years.

- Type 2: *Clustered absolute errors.* This type of outliers refers to a continuous sequence of Type 1 outliers. The length of this sequence can be determined depending on the application, where the sample interval and type of sensor data should be considered.

- Type 3: *Random errors.* A more usual type of outliers is indicated by observations not falling within the threshold with the normal behavior of data. These random errors displaying values inconsistent with normal behavior of data last for a very short period of time.

- Type 4: *Long-Term errors.* This type of outliers refers to a continuous sequence of Type 3 outliers. The length of this sequence can be determined depending on the application, considering the sample interval of the sensor observations and type of sensor data.

Events are a special type of outliers that occur at nodes in the neighborhood at the same time. This type of outliers can be determined using the criteria for Type 4 outliers combined with specific minimum number of nodes displaying the same behavior. This number depends on the type of sensor data.

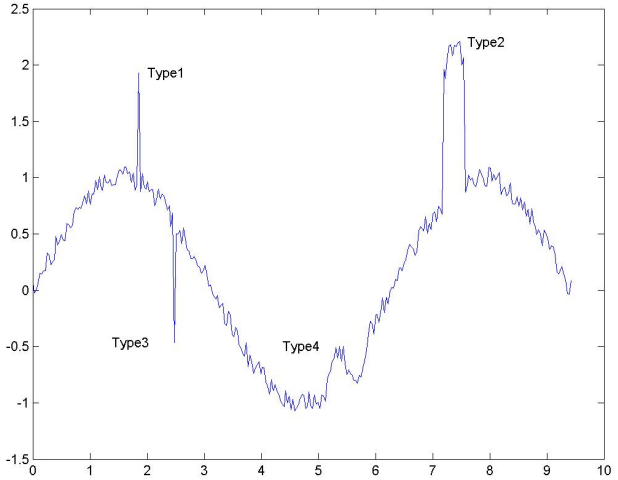


Figure 3.1: Examples of different types of outliers

3.3 Sensor Dataset

To compare the four different labelling techniques, we need a number of different datasets. We are interested in datasets containing a different number of features for a different number of sensor nodes and days. We also need a dataset, in which the correlation between sensor nodes varies. A dataset matching our description is the Grand St. Bernard dataset [108].

The Grand St. Bernard dataset has been collected by a multi-hop wireless sensor network, deployed at the Grand St. Bernard pass, located between Switzerland and Italy. The setup consists of 23 sensor nodes. These nodes measure, among other meteorological characteristics of the environment, the temperature and humidity during a period of two months [108] with the sampling frequency of two minutes. The nodes are deployed in two clusters. The small cluster consists of 5 nodes, while the big cluster consists of 18 nodes. Each cluster has a base station. Figure 3.2 depicts the topology of the network.

Even though the nodes are relatively close to each other, for instance, the distance between node 9 and node 18 is approximately 0.764 km, the correlation between the sensor observations is not necessary strong. Figure 3.3 illustrates the mean correlation for humidity and temperature for both clusters. Humidity

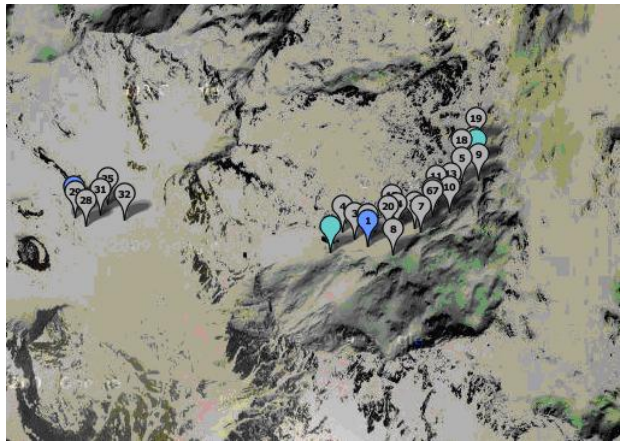


Figure 3.2: Wireless sensor network located in the Grand St. Bernard Pass, Switzerland [108]

Days	09/26	09/27	09/28	09/29	09/30	10/01	10/02	10/03	10/04	10/05	10/06	10/07	10/08	10/09	10/10	mean
Min	-0.95	-0.98	-0.92	-0.20	-0.89	-0.72	-0.39	-0.52	-0.47	-0.61	-0.91	-0.68	0.00	-0.32	-0.33	-0.59
Max	0.96	0.98	0.97	0.99	0.96	0.94	0.94	0.99	0.97	0.98	0.98	0.97	0.97	0.96	0.96	0.97
Std	0.48	0.48	0.43	0.43	0.46	0.41	0.30	0.40	0.37	0.47	0.44	0.43	0.34	0.33	0.38	0.41
Mean	0.08	0.07	0.10	0.53	0.18	0.23	0.38	0.60	0.47	0.64	0.16	0.31	0.55	0.52	0.35	0.34
|Mean|	0.37	0.36	0.33	0.54	0.37	0.34	0.40	0.66	0.53	0.77	0.37	0.43	0.55	0.54	0.40	0.46

Table 3.1: Humidity correlation for each day

correlation for each day is depicted in Figure 3.4 and Table 3.1. Temperature correlation for each day is depicted in Figure 3.5 and Table 3.2. In these tables the mean correlation for temperature and humidity lies around 0.34 and 0.42 respectively. The standard deviation is very high. This means that the correlation varies strongly for individual nodes. It can be clearly seen that humidity correlation between nodes of the small cluster is stronger than their temperature correlation. This is the other way around, however, for the big cluster. It means that the temperature correlation between nodes of the big cluster is stronger than their humidity correlation.

Another observation is that there is a relatively weak correlation between humidity and temperature of nodes of the two clusters. The correlation between humidity and temperature on each node is also low (this is displayed using the color of the nodes). This correlation between humidity and temperature is also depicted for each day separately in Figure 3.6 and Table 3.3. The standard devi-

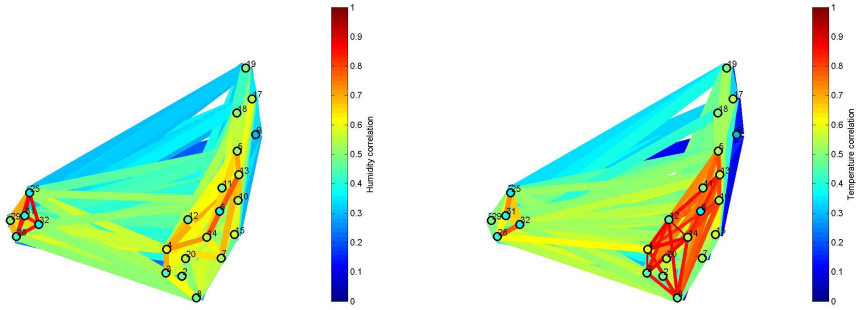


Figure 3.3: Mean correlation of humidity and temperature during 15 days period (2007/09/26 - 2007/10/10)

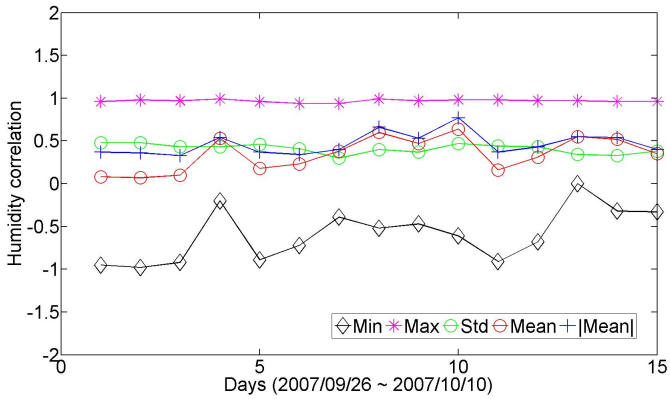


Figure 3.4: Humidity correlation for each day

Days	09/26	09/27	09/28	09/29	09/30	10/01	10/02	10/03	10/04	10/05	10/06	10/07	10/08	10/09	10/10	mean
Min	-0.58	-0.79	-0.85	0.00	0.00	0.00	-0.72	-0.27	-0.14	-0.23	-0.61	-0.74	-0.50	-0.57	-0.56	-0.44
Max	1.00	0.99	0.99	0.99	1.00	0.98	0.99	0.99	1.00	0.99	1.00	0.98	0.99	0.98	0.98	0.99
Std	0.44	0.52	0.50	0.39	0.33	0.44	0.33	0.37	0.35	0.37	0.42	0.49	0.45	0.46	0.37	0.42
Mean	0.40	0.56	0.60	0.65	0.30	0.55	0.20	0.55	0.39	0.48	0.17	0.29	0.38	0.37	0.33	0.42
|Mean|	0.45	0.66	0.72	0.65	0.30	0.55	0.26	0.58	0.40	0.51	0.32	0.46	0.44	0.45	0.35	0.47

Table 3.2: Temperature correlation for each day

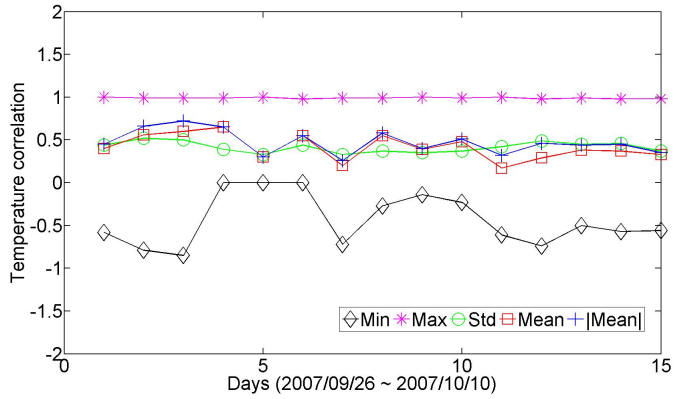


Figure 3.5: Temperature correlation for each day

Days	09/26	09/27	09/28	09/29	09/30	10/01	10/02	10/03	10/04	10/05	10/06	10/07	10/08	10/09	10/10	mean
Min	-0.86	-0.99	-0.89	-0.93	-0.91	-0.69	0.03	-0.84	-0.60	-0.75	-0.24	-0.82	-0.94	-0.82	-0.84	-0.74
Max	0.99	0.99	0.99	0.68	0.46	0.85	0.99	0.72	0.72	0.69	0.97	0.96	0.86	0.94	0.87	0.85
Std	0.54	0.63	0.55	0.53	0.30	0.39	0.31	0.67	0.37	0.46	0.35	0.54	0.66	0.49	0.52	0.49
Mean	0.20	0.18	0.02	-0.56	-0.09	-0.36	0.51	-0.27	-0.20	-0.34	0.33	-0.15	-0.38	-0.20	-0.08	-0.09
|Mean|	0.47	0.54	0.45	0.69	0.21	0.45	0.51	0.69	0.34	0.52	0.41	0.44	0.66	0.44	0.43	0.48

Table 3.3: Correlation between humidity and temperature for each day

ation, minimum and maximum values show that the correlation is very unstable. The overall correlation between humidity and temperature is zero. The row displaying the $|mean|$ (the mean of the absolute value of the correlation) shows that the correlation is around 0.5. Overall the correlation depends on the distance between the nodes. However, the correlation between humidity and temperature for different nodes varies day by day. Table 3.4 illustrates the correlation between humidity, temperature and distance.

Figure 3.7 (left) illustrates that the correlation for humidity is overall very weak for one day. In the big cluster there are strong correlations spanned over relatively large distances, while most nodes, located very close to each other, have a weak correlation. The temperature correlation plot in Figure 3.7 (right) illustrates that there is a very strong correlation between nodes of the two clusters for the same day. Figure 3.8 illustrates the correlation plot for the humidity and temperature for another day. It can be clearly seen that distance and humidity are correlated, while temperature and distance are not. This shows that in this

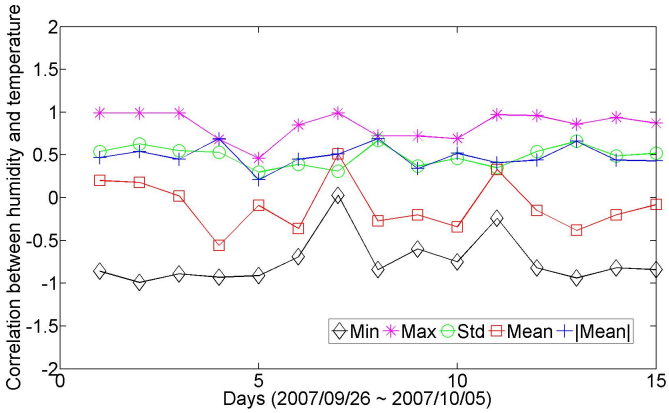


Figure 3.6: Correlation between humidity and temperature for each day

Correlation	Humidity	Temperature	Distance
Humidity	1	-0.09	-0.37
Temperature	-0.09	1	-0.30
Distance	-0.37	-0.30	1

Table 3.4: Correlation between humidity, temperature and distance

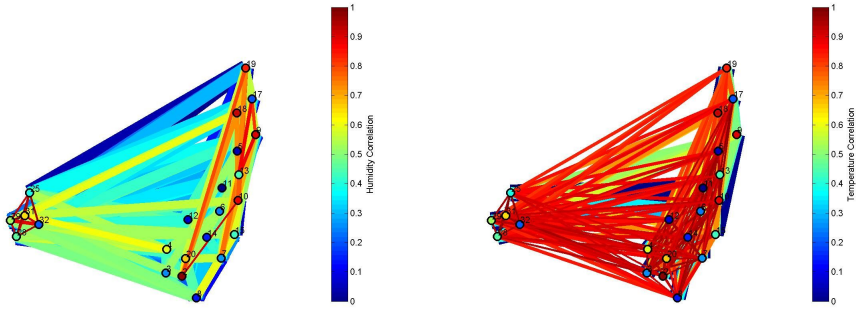


Figure 3.7: Correlation of humidity and temperature on 2007/09/28

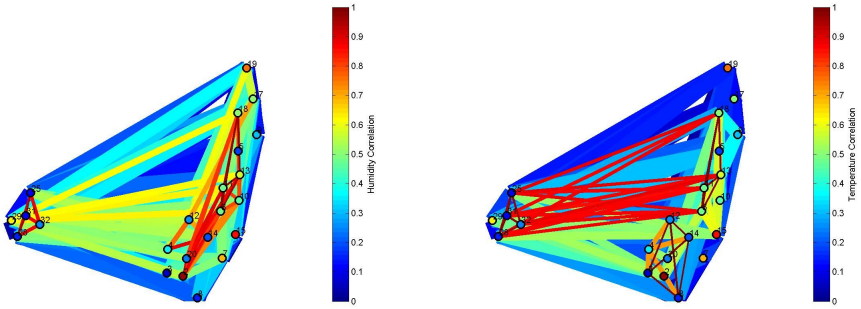


Figure 3.8: Correlation of humidity and temperature on 2007/10/06

Features	Days per set	Nodes	Nodes per set	Correlation	Datasets
h,t,h+t	1	18	1	-	329
h,t,h+t	1-2	18	1-5	0.7	34
h,t,h+t	1	5	1	0.7	2
h,t,h+t	1	18	18	-	15
h,t,h+t	1	5	5	-	15

Table 3.5: Different subdatasets, h:humidity, t:temperature, h+t: humidity and temperature combined

dataset, obviously no systematic trend exists and moreover no unified assumptions regarding correlation between humidity and temperature can be made.

From the Grand St. Bernard dataset, we select various subdatasets having different characteristics in terms of number of days, number of nodes, number of features, and correlation values between humidity and temperature. Table 3.5 provides an overview about these different datasets. As it can be seen from the table, we consider the correlation between humidity and temperature as well as the correlation between the nodes for two datasets. In these two datasets, the minimum correlation value is 0.7. Through extensive experiments, we realize that no subdataset could be found to match a bigger correlation value. For the other three datasets reported in the table, we have made a combination of days and nodes. We separate the two clusters (containing 5 and 18 nodes) because the distance between the clusters is large compared to the distances between

the nodes inside the clusters. This has been done on the assumption that the correlation between the nodes depends on the distance.

3.4 Data Labelling Techniques

To label the data, we use four different techniques. We first contextualize these techniques, explain how we determine certain critical parameters, and then indicate complexity of these techniques. Furthermore, we illustrate the characteristics of these techniques and show their advantages and disadvantages.

3.4.1 Mahalanobis Distance-Based Labelling Technique

Distance-based techniques identify outliers based on the measure of full dimensional distance between a point and its nearest neighbor in the dataset. Euclidean distance and Mahalanobis distance are commonly used as *similarity measures* in distance-based techniques. Distance-based techniques consider points that are distant from other data points in the dataset as outliers.

Euclidean distance is defined as:

$$ED_i = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.1}$$

where x_i, y_i represent two data points and n is dimensionality of the data.

Labelling data based on Euclidean distance has the advantage of having a low computational complexity. It, however, has the disadvantage that it tries to fit the data into a circular shape. Temperature and humidity data of the Grand St. Bernard dataset, however, do not often present themselves in a circular shape. Applying that labelling technique on this dataset will therefore result in a high number of data being falsely labelled as outliers.

Mahalanobis distance is an extension to Euclidean distance, in the sense that it takes the covariance matrix into consideration. For a d-dimensional multivariate sample, such as $\{x_i : i = 1, 2, \ldots n\}$, Mahalanobis distance is defined as:

$$MD_i(x) = \sqrt{(x - \mu)\Sigma^{-1}(x - \mu)^T} \tag{3.2}$$

where Σ represents the $d \times d$ covariance matrix and μ is the multivariate mean.

Using Mahalanobis distance to label the data, an outlier is considered to be a point whose Mahalanobis distance is larger than a certain threshold. We define

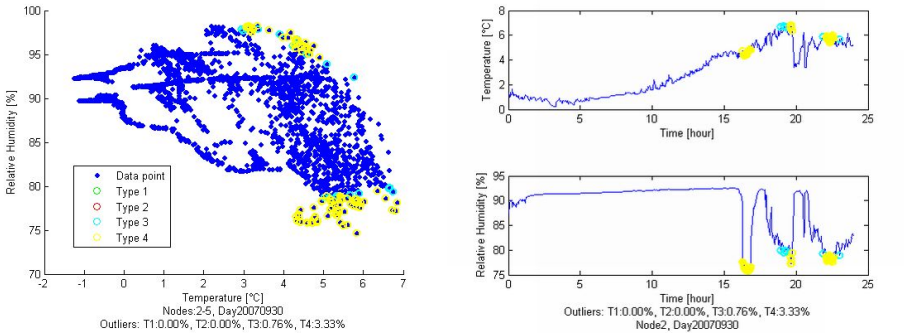


Figure 3.9: Outliers found using Mahalanobis distance-based labelling technique

this threshold value as the average value between the minimum and the maximum value of the Mahalanobis distance values assigned to the data points.

A main advantage of the Mahalanobis distance-based labelling technique compared with the Euclidean distance is that it is applicable for both elliptical and circular shaped data. To calculate the Mahalanobis distance for the points in a dataset, the covariance matrix needs to be specified. This makes the computation of Mahalanobis distance more expensive.

The requirement to use the Mahalanobis distance is that the dataset should form at most one cluster. Furthermore the cluster should have a circular, linear, or elliptical shape. These requirements are not always met by the humidity and temperature data we use. Often, the data has more than one cluster or had an L- or O- like shape. An example of a dataset labelled using Mahalanobis distance-based technique is depicted in Figure 3.9 .

3.4.2 Density-Based Labelling Technique

Density-based techniques take the local density into account when searching for outliers. These techniques can effectively identify local outliers in datasets with diverse clusters.

We implement a simple density-based labelling technique using a histogram. The definition of an outlier according to this technique is that a data point is an outlier if it resides in a pixel of a grid whose density is lower than a certain threshold. For 2-D datasets, we use a *bivariate* histogram. The data range for each feature is divided in a number of pixel. Then the number of samples in each pixel is counted. If there are less samples in a pixel than the defined threshold,

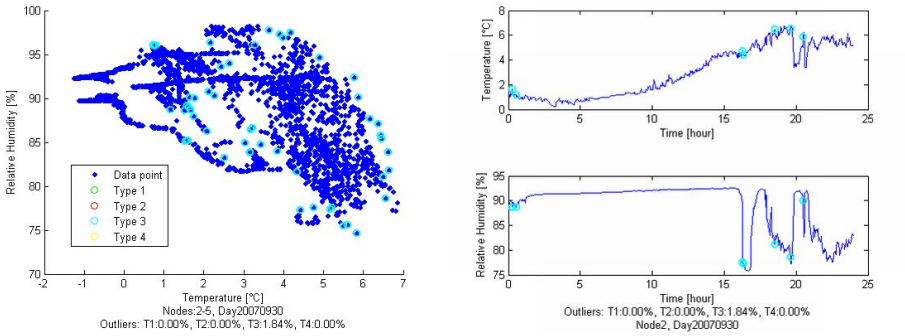


Figure 3.10: Outliers found using density-based labelling technique

samples in the pixel are considered as outliers.

The number of pixels is fixed. We use twenty pixels for each feature. To separate the outliers from the normal data, we use a threshold, which is determined using a fixed percentage of the density values. To calculate the density for 1-D datasets, we use a regular histogram and apply the same procedure.

When evaluating this procedure with regard to temperature and humidity values, we have to consider how these time-dependent measurements translate into a 2-D plot of two features. Since the sampling frequency of the observations is 2 minutes, if the temperature or humidity changes fast, the 2-D plot will show a line of dots with a big distance between the dots. When the temperature or humidity values change gradually, the 2-D plot will show a connected line. Also, when the values change fast following a path in the 2-D space that is already formed by previous values, the current values will fall in an area with a higher density. By doing so, only unique sudden changes in temperature and humidity values are labelled as outliers. An example of a dataset labelled using this density-based technique is depicted in Figure 3.10.

3.4.3 Running Average-Based Labelling Technique

We base our running average approach on techniques used for streaming datasets. General outlier detection techniques work well in static datasets, in which all data points are stationary. However, in streaming and dynamic datasets, a large volume of data is continuously and fast transferred in an ordered sequence, and also data may be constantly added, removed, or updated. In this dataset, a data model built in a particular time instant may be invalid in consequent time

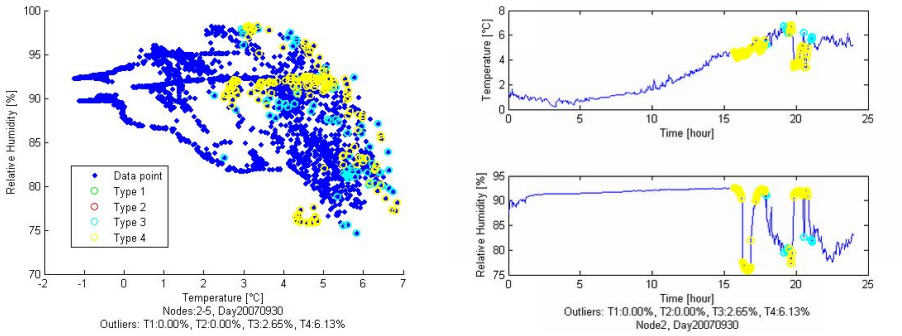


Figure 3.11: Outliers found using running average-based labelling technique

instants. Thus, data stream can be viewed as an infinite process consisting of data which continuously evolves with the time.

A running average-based technique is a *filter*, that is often used to smoothen time series data. This filter calculates the mean value for a fixed number of samples. This means that *running average* for a W number of samples S is calculated as:

$$R_n = \frac{s_{n-W/2} + s_{n-W/2+1} \cdots s_{n+W/2-1} + s_{n+W/2}}{W} \tag{3.3}$$

An outlier is defined by taking the absolute value of the difference between the original values and the values calculated by applying the running average filter. Each value above the threshold value C, calculated as $C = median + 2 * std$ is considered as an outlier. Using the median instead of the mean to calculate C, is to minimize the influence of outliers.

The outliers defined by this technique do not dependent on the range of the values in the dataset, but purely depend on the surrounding samples. Direct implication of this is that a collection of extreme values like a heat wave would not be labelled as outliers using the running average labelling technique. An example of a dataset labelled using this running average-based technique is depicted in Figure 3.11.

3.4.4 Naive Bayesian-Based Labelling Technique

We choose the Naive Bayesian classifier presented by [28] to label the data. To be able to use Bayesian classifiers, the data should be divided into a number of

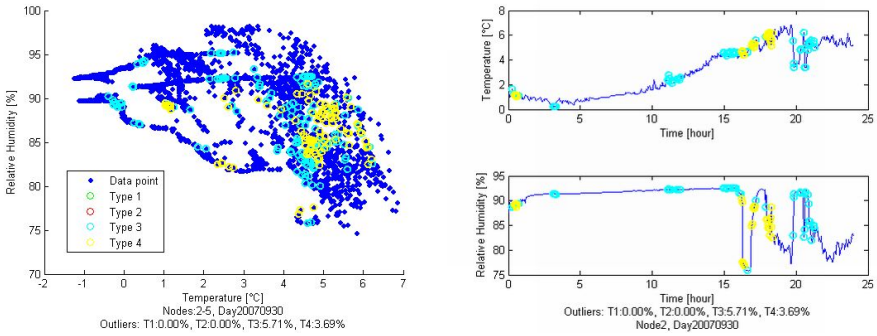


Figure 3.12: Outliers found using Naive Bayesian-based labelling technique

mutually exclusive classes over the full length of the data range. This technique needs to learn the data distribution before it can detect outliers. During the training phase, for each node, number of occurrences of each combination of historical and current samples of the node itself as well as current samples of the nodes neighbors is calculated. Based on these numbers, the labelling technique calculates the probabilities of the occurrence of different combinations of classes for the current, previous and neighbor samples [28]. Since, this technique calculates for each sample its probability to be in each possible class, if a sensor value does not fit in the class with the highest probability, it is labelled as an outlier. This definition of outlier, however, only applies to 1-D datasets.

To be able to use the technique for 2-D datasets, we consider an observation to be an outlier if one or both sensor values are marked as outlier by the technique. In the original paper the local neighborhood consists of two randomly chosen distinct neighbors from the immediate one-hop neighborhood of the computing node. Since we do not know the connectivity of the Grand St. Bernard deployment, we use two randomly chosen neighbors within a radius of 0.5 kilometers around the computing node. We choose to use four classes for each feature. The boundaries of the classes are defined as:

$$[min(\vec{s_d}), median(\vec{s_d}) - std(\vec{s_d}), median(\vec{s_d}), median(\vec{s_d}) + std(\vec{s_d}), max(\vec{s_d}))] \tag{3.4}$$

where $\vec{s_d}$ is a vector containing sensor data of feature d. An example of a dataset labelled using this technique is depicted in Figure 3.12.

Implementation of the original algorithm presented in [28] did not lead to good outlier detection results. As it can be observed from Figure 3.13, this technique

Temperature	-0.92	0.22	2.79	5.37	10.93
Humidity	64.75	79.34	84.86	90.39	93.66

Table 3.6: Four classes used in the Naive Bayesian-based labelling technique

labels around 37% of the data as outliers. It seems that the data values labelled as outliers are mostly concentrated on the boundaries of the classes. One of the problems causing this is that the Grand St. Bernard dataset is not linearly repeatable. Another problem is that the dataset is very diverse both in terms of the range and distribution of the data. Moreover, at some periods of time the data barely changed, while at other times the data kept fluctuating constantly.

We improve the percentage of detected outliers in the original algorithm by changing the distinguishing factor of the classes to:

$$p_n \times c < p_s \tag{3.5}$$

where p_n and p_s are the probability for sensor observations to be in class n and s, respectively, and c is a constant value representing the weight of the probability p_n. In this way, the probability of class p_n to be relevant should be c times higher. We repeat the experiments using the new distinguishing factor for different values of c ($c = 1, 10, 100, 100$). Figure 3.13, 3.14, 3.15, and 3.16, illustrate these experimental results. The four classes used for this dataset are shown in Table 3.6.

To address the problem of having non-linear repeatable data, we convert the cartesian coordinates to polar coordinates, by which the performance of the algorithm got better. However, the translation itself is very difficult. The reason behind this is that this translation is usually easy for data in a circular or elliptical shape. However, the datasets selected from the Grand St. Bernard have irregular shapes with no similarities between them. This makes it hard to prepare the data for labelling using this algorithm. To solve the problem of irregularity we can apply a clustering algorithm to determine the centers of the data, which help easier translation into polar coordinates.

Another problem faced by this technique, is the correlation between the nodes. This correlation is not necessary constant and is also not always bound to distance. Therefore, it is not always the case that the sensor values used by neighboring nodes to calculate the probability are related to sensor values of the node in question. So, the prediction based on the neighbor node might not help to predict the current sensor observation and might even disturb the prediction. An example of this correlation changes is depicted in Figures 3.17 and 3.18, which

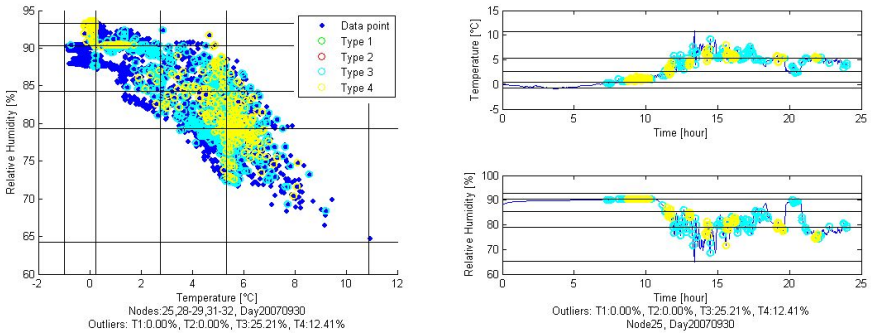


Figure 3.13: Data labelled using Bayesian classifiers, c=1

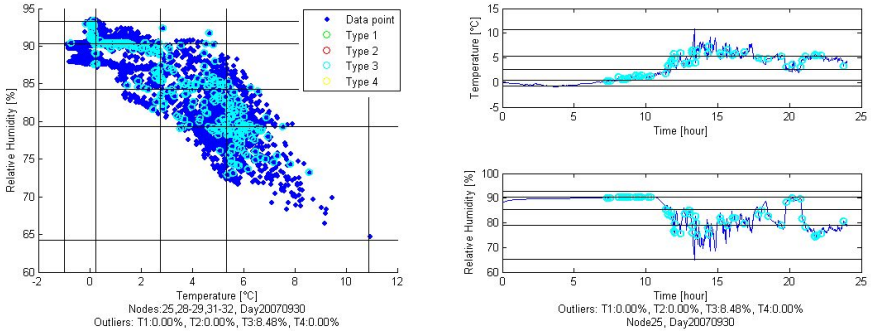


Figure 3.14: Data labelled using Bayesian classifiers, c=10

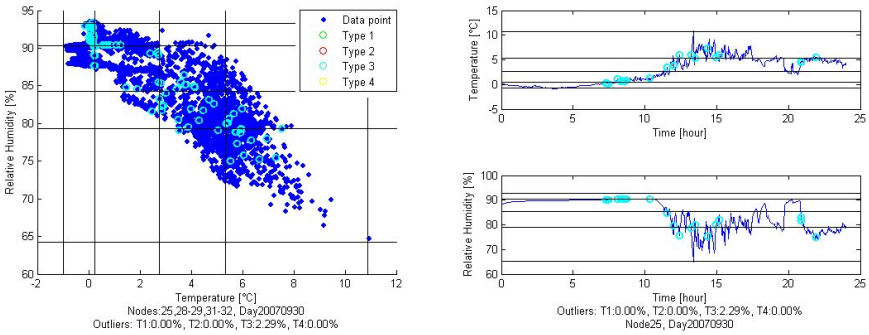


Figure 3.15: Data labelled using Bayesian classifiers, c=100

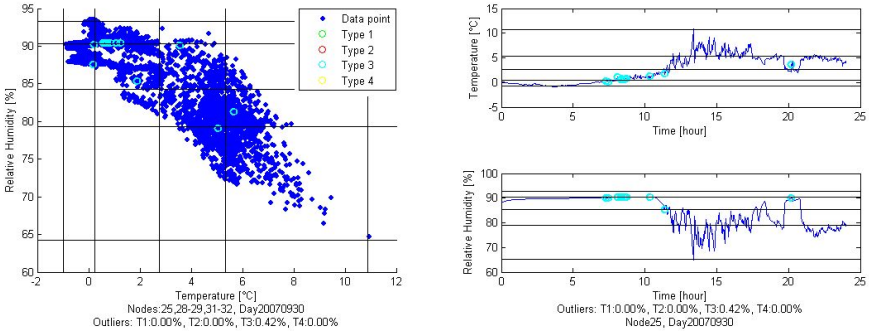


Figure 3.16: Data labelled using Bayesian classifiers, c=1000

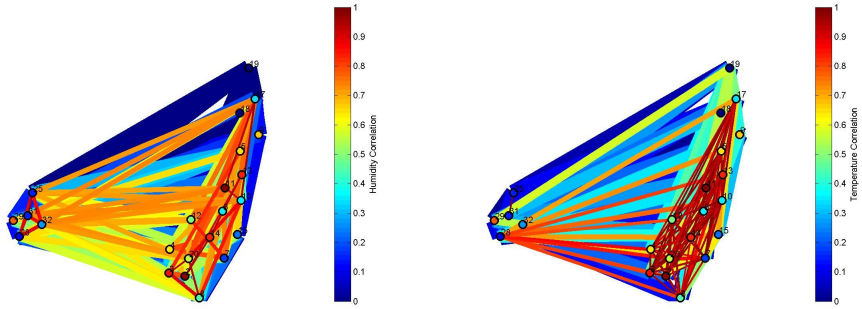


Figure 3.17: Correlation between humidity and temperature on 2007/09/26

show the correlation between humidity and temperature in both small and big network clusters for two consecutive days.

The complexity of this Naive Bayesian-based labelling technique is $O(n)$ for training, $O(c^3)$ for the storage of the probability tables, and $O(n)$ for the outlier detection procedure, where n is number of classes and c is a constant value representing the weight of the probability of a data point fitting in class n.

Due to poor results we got during the experiments, we exclude the Naive Bayesian-based labelling technique in the rest of the discussion.

3.5 Comparison

Since there is no objective way to verify the results of the three aforementioned labelling techniques, i.e., Mahalanobis distance-based, density-based, and running average-based, we will focus on the characteristics of these techniques. Outliers of Type 1 and 2 are not often found in the Grand St. Bernard dataset. We will show two examples of datasets with extreme values and four datasets with specific shapes. We will show the performance of the three techniques on labelling these datasets. Additionally, we compare these labelling techniques in terms of their computational complexity.

3.5.1 Performance Comparison based on Datashapes

The shape of the dataset is an important factor in deciding which labelling technique to choose to label a dataset. To illustrate this, we use six dataset examples

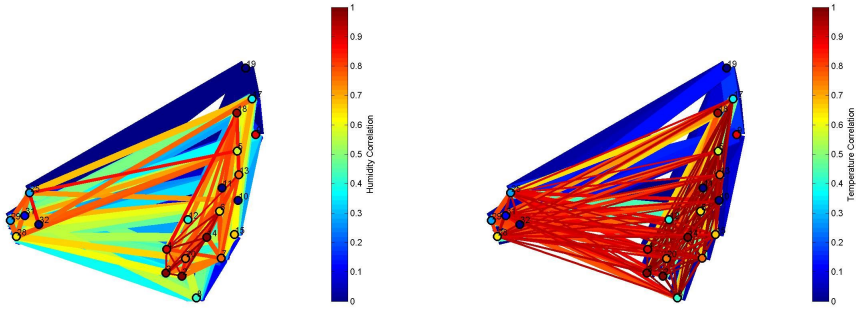


Figure 3.18: Correlation between humidity and temperature on 2007/09/27

of different shapes and present the performance evaluation of the three different techniques applied on these datasets. We will show two datasets containing extreme values and present four shapes, that are the *ring*, *ellipse*, *curved*, and an *irregular shape*. The reason behind our choice is that extreme values do not occur very often, and these shapes are very common for the temperature and humidity readings in the Grand St. Bernard dataset. For each shape, three sets of plots are presented. Each set contains a 2-D plot of labelled temperature and humidity data and two time plots for temperature and humidity individually.

We show the four types of outliers in each set. To determine the type of outlier, we choose $-5\,^{\circ}\mathrm{C}$ and $25\,^{\circ}\mathrm{C}$ as the minimum and maximum temperature. For humidity we use 20% and 100% as minimum and maximum values. These values are based on the temperature and humidity data of the Federal Office of Meteorology and Climatology of Switzerland [83] for the months September and October. To distinguish between outliers of different types, we choose the minimum length of a sequence to be four samples. This translates to a time period of 8 minutes.

Ring Shape

In Figures 3.19, 3.20, and 3.21, a dataset is depicted that has a ring shape with a hole in the middle. In this dataset, only outliers of Type 3 and 4 are present. Samples which are intuitively identified as outliers can mainly be found on the outer and inner bounds of the ring. The Mahalanobis distance-based labelling technique (shown in Figure 3.19), only identifies outliers of Type 3 and 4 outside the ring, while the density-based labelling technique (shown in Figure 3.20)

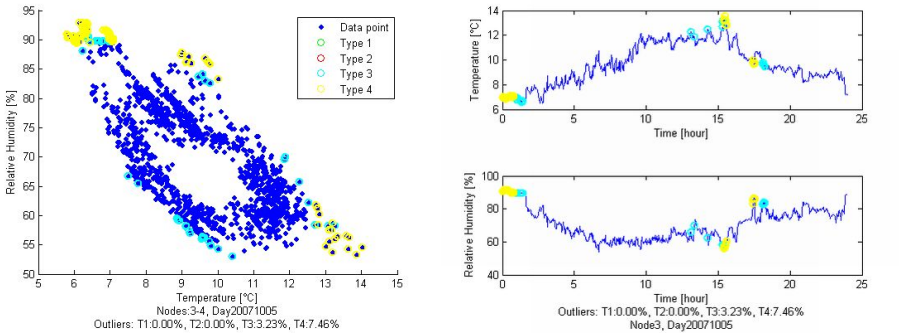


Figure 3.19: Performance of Mahalanobis distance-based labelling technique for ring datashape

identifies outliers of Type 3, being both inside and outside the ring. The running average-based labelling technique (shown in Figure 3.21), however, finds outliers of both types inside and outside the ring.

Ellipse Shape

In Figures 3.22, 3.23, and 3.24, a dataset is depicted that has an ellipse shape. In this dataset, only outliers of Type 3 and 4 are present. Intuitively, outliers can be found at the edge of the dataset and in the gaps of these edges, and might also occur in the humidity-time graph between 6am and 14am. The Mahalanobis distance-based labelling technique (shown in Figure 3.22), only identifies outliers of Type 3 and 4 that are only found on the outer edges and not in the gaps. The density-based labelling technique (shown in Figure 3.23) mainly identifies outliers Type 3 and two clusters of Type 4. These outliers are both on the edge of the ellipse as well as in the gaps of the ellipse and even inside the ellipse. The running average-based labelling technique (shown in Figure 3.24), however, finds outliers of both types, both inside and outside the edges of the ellipse. An important observation is that when the density in the middle of the ellipse is high and there are no other small clusters outside the ellipse, the density-based and Mahalanobis distance-based labelling techniques will identify the same outliers.

Curved Shape

In Figures 3.25, 3.26, and 3.27, a curved dataset is depicted. Datasets having a strong correlation between humidity and temperature and gradual change of

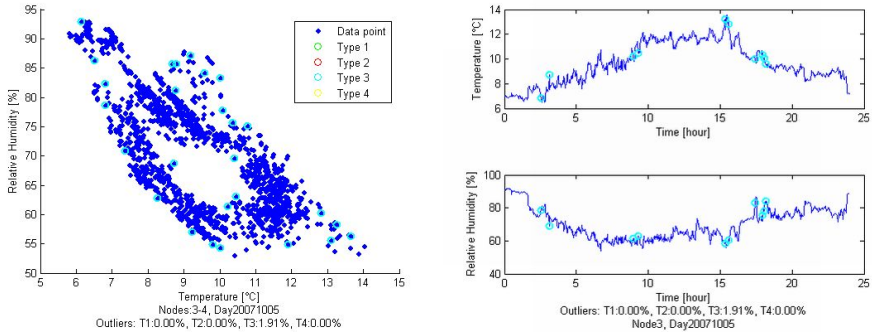


Figure 3.20: Performance of density-based labelling technique for ring datashape

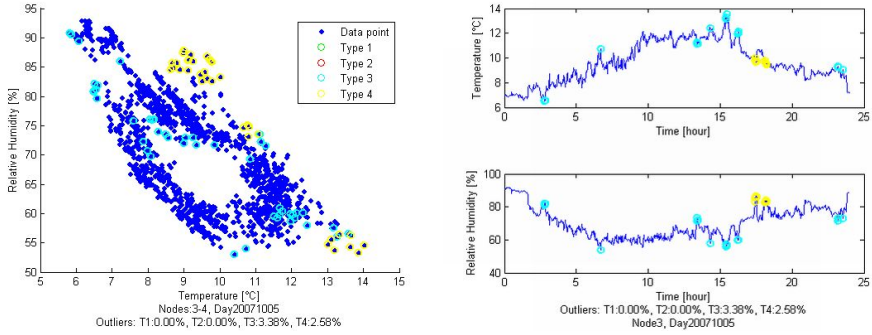


Figure 3.21: Performance of running average-based labelling technique for ring datashape

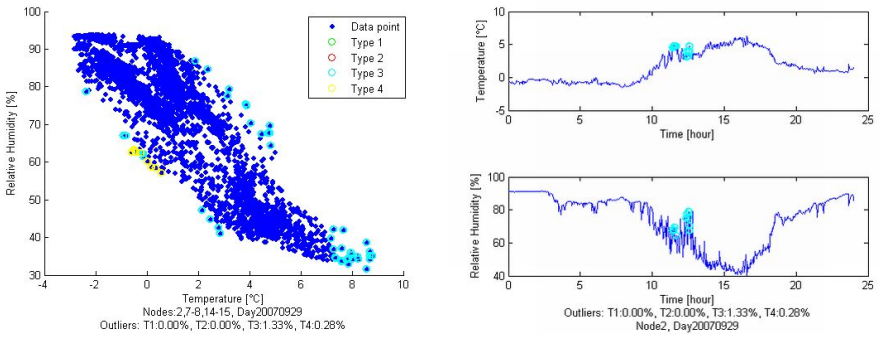


Figure 3.22: Performance of Mahalanobis distance-based labelling technique for ellipse datashape

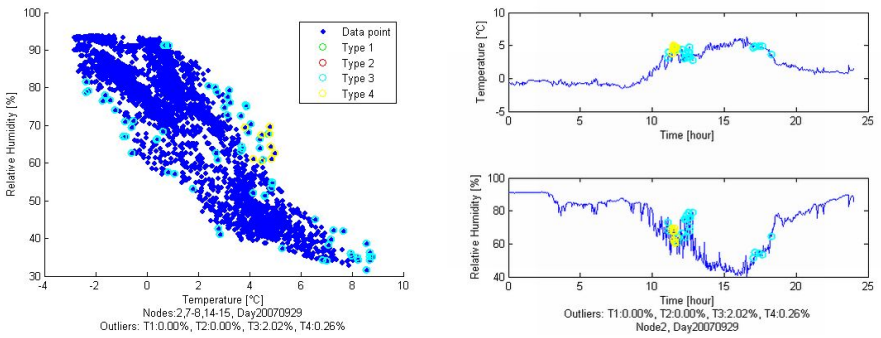


Figure 3.23: Performance of density-based labelling technique for ellipse datashape

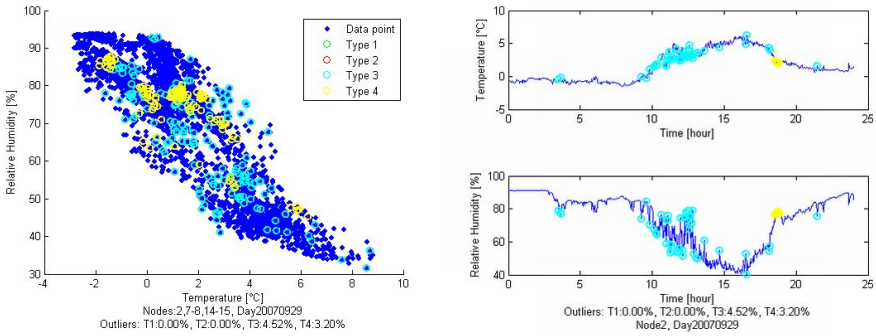


Figure 3.24: Performance of running average-based labelling technique for ellipse datashape

temperature and humidity over time have a linear or curved shape. Due to the fact that the shapes are very regular as shown in these figures, thus it is discussable which values are outlier, and even whether this dataset contains any outliers. Possible outliers of Type 3 and 4 can be found at both ends of the curve and at the edge of the curve.

The Mahalanobis distance-based labelling technique (shown in Figure 3.25) identifies outliers of Type 3 and 4, while the density-based labelling technique (shown in Figure 3.26) identifies only outliers of Type 3. The running average-based labelling technique (shown in Figure 3.27), however, finds outliers of both types.

Irregular Shape

In Figures 3.28, 3.29, and 3.30, an irregular dataset is depicted. This shape has a number of vaguely identifiable clusters. Between those clusters, there are a lot of non-connected readings. We observe that readings that would intuitively be pointed out as outliers, can be found between those clusters and in the very small cluster below the three bigger clusters.

The Mahalanobis distance-based labelling technique (shown in Figure 3.28) identifies outliers of Type 3 and 4. It defines all the sensor observations in the small cluster as Type 4 outliers. It is clear that this technique considers the data as one cluster and assumes that the outliers occur only outside the three clusters. The density-based labelling technique (shown in Figure 3.29) only identifies outliers of Type 3, both between and around the three clusters. The running

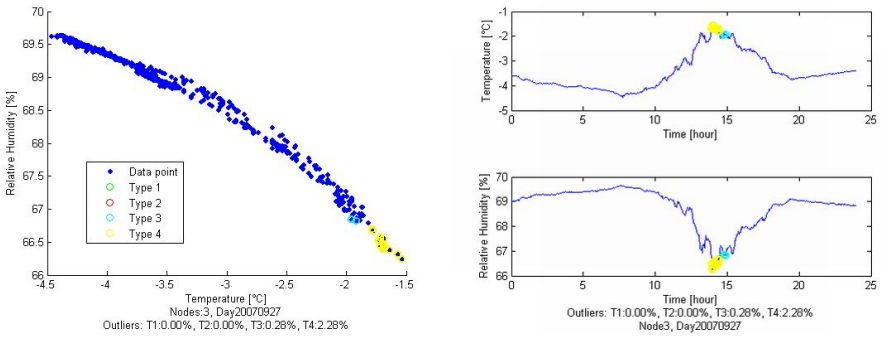


Figure 3.25: Performance of Mahalanobis distance-based labelling technique for curved datashape

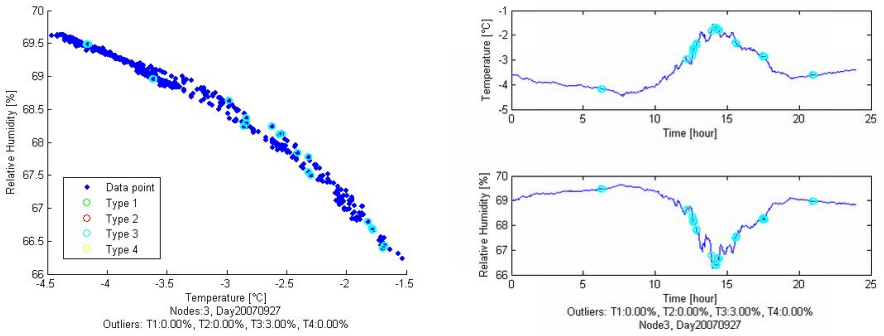


Figure 3.26: Performance of density-based labelling technique for curved datashape

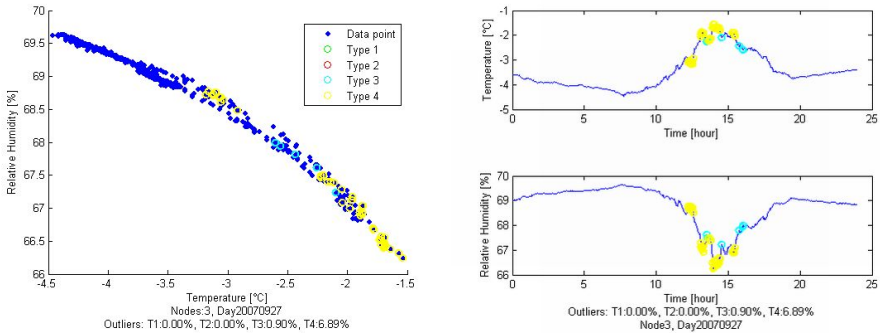


Figure 3.27: Performance of running average-based labelling technique for curved datashape

average-based labelling technique (shown in Figure 3.30), however, finds outliers of both types, inside and outside the three clusters.

Datasets with Extreme Values

In Figures 3.31, 3.32, and 3.33, a dataset with extreme values is depicted. The boundaries for Type 1 and 2 outliers are set at $-5\,°C$ - $25\,°C$ for the temperature, and at 20% and 100% for humidity. Possible outliers of Type 1 and 2 can be found at temperature more than $25\,°C$, while outliers of Type 3 or 4 can be found at the edges of the shape on the left side of the graph.

The Mahalanobis distance-based labelling technique finds a part of the values at the right side of the $25\,°C$ boundary (shown in Figure 3.31). It does not find outliers of Type 3 or 4. The density-based labelling technique (shown in Figure 3.32) identifies all possible Type 1 or 2 outliers. It also finds some outliers of Type 3. The running average-based labelling technique (shown in Figure 3.33), however, finds outliers of all four types.

In Figures 3.34, 3.35, and 3.36, another dataset with extreme values is depicted. The boundaries for Type 1 and 2 outliers are set at $25\,°C$ for the temperature, and at 20% and 100% for humidity. Possible outliers of Type 1 and 2 can be found at the left side of $-5\,°C$ and below 20% humidity, while outliers of Type 3 or 4 can be found in the two small clusters near the big cluster in the upper right corner and even at the edges of the big cluster.

The Mahalanobis distance-based labelling technique (shown in Figure 3.34) only identifies the samples in the cluster in the left upper as outliers. This happens

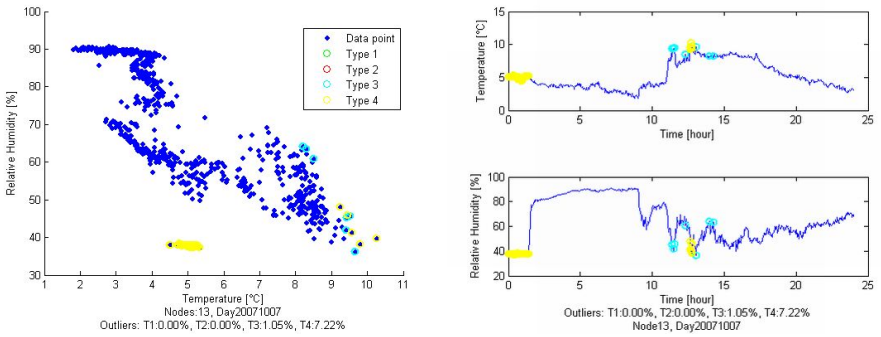


Figure 3.28: Performance of Mahalanobis distance-based labelling technique for irregular datashape

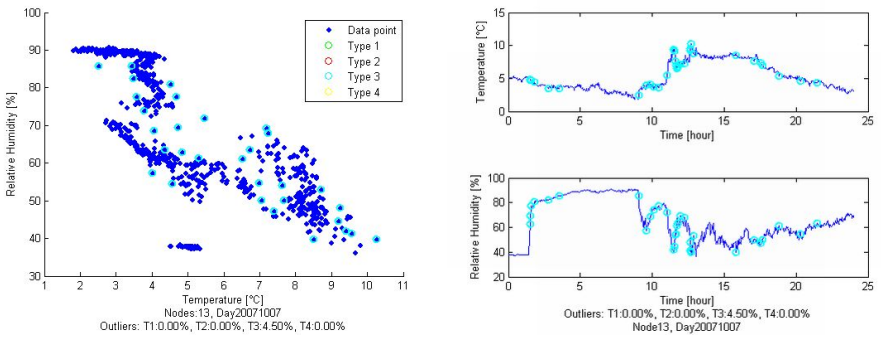


Figure 3.29: Performance of density-based labelling technique for irregular datashape

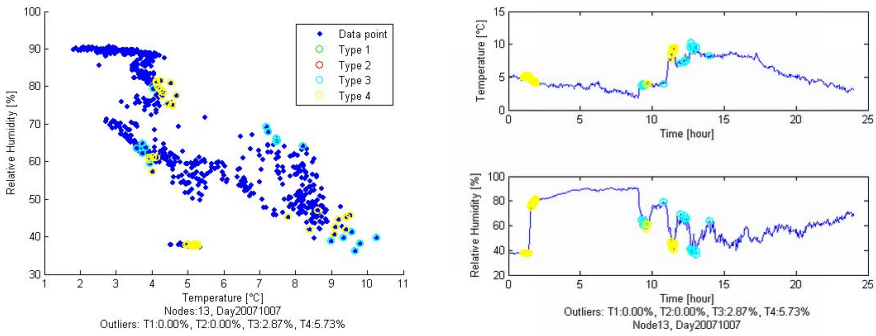


Figure 3.30: Performance of running average-based labelling technique for irregular datashape

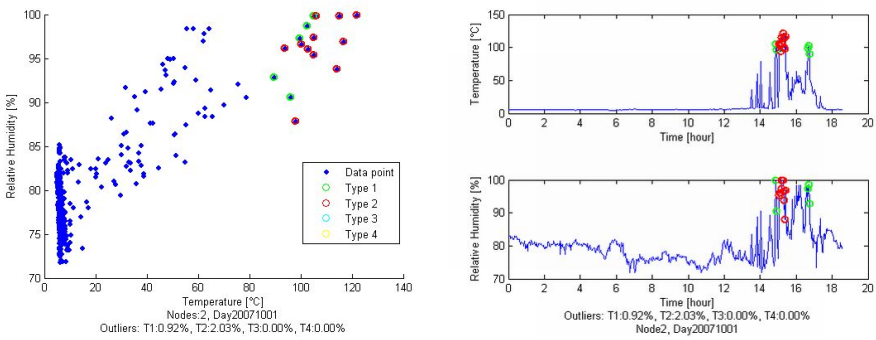


Figure 3.31: Performance of Mahalanobis distance-based labelling technique for extreme values

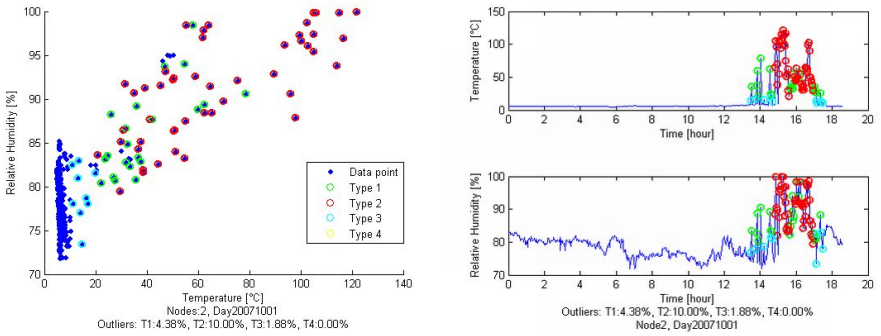


Figure 3.32: Performance of density-based labelling technique for extreme values

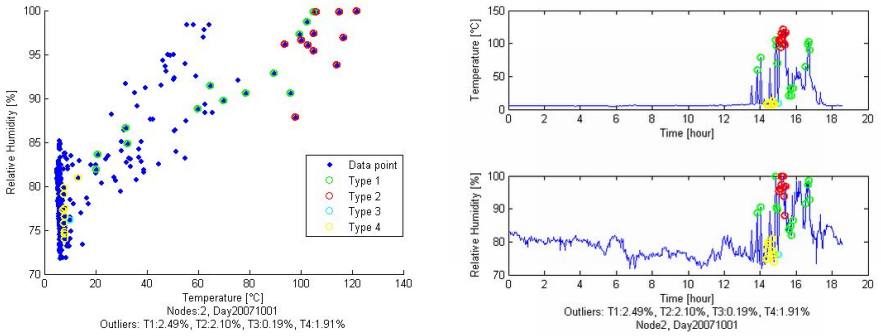


Figure 3.33: Performance of running average-based labelling technique for extreme values

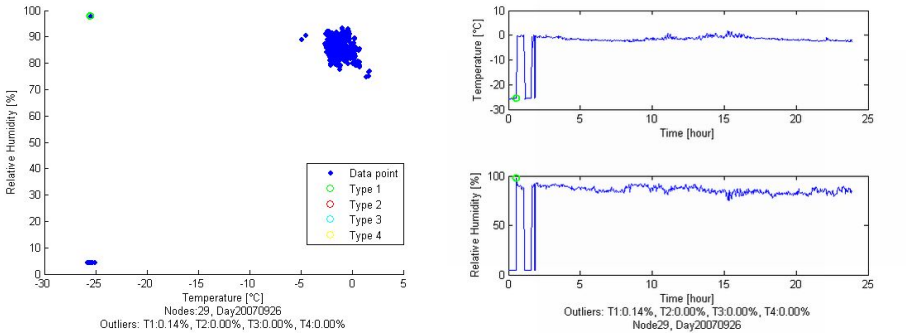


Figure 3.34: Performance of Mahalanobis distance-based labelling technique for extreme values

because of the calculation of this distance, the choice of the threshold and the relatively large amount of samples in the left lower corner. The Mahalanobis distance-based labelling technique does not find outliers of Type 3 or 4. The density-based labelling technique (shown in Figure 3.35) also misses the cluster in the left lower corner because of the relative high density in this cluster. It does identify the cluster in the left upper corner and it also finds some outliers around the big cluster. The running average-based labelling technique (shown in Figure 3.36), however, identifies both clusters at the left side of the graph. It also finds some outliers around the big cluster.

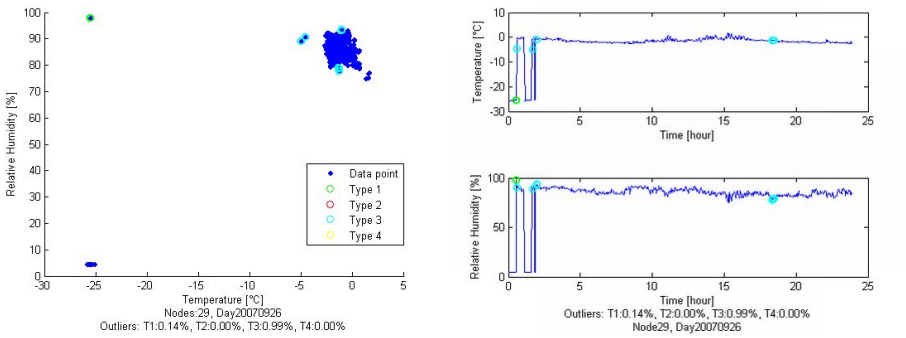


Figure 3.35: Performance of density-based labelling technique for extreme values

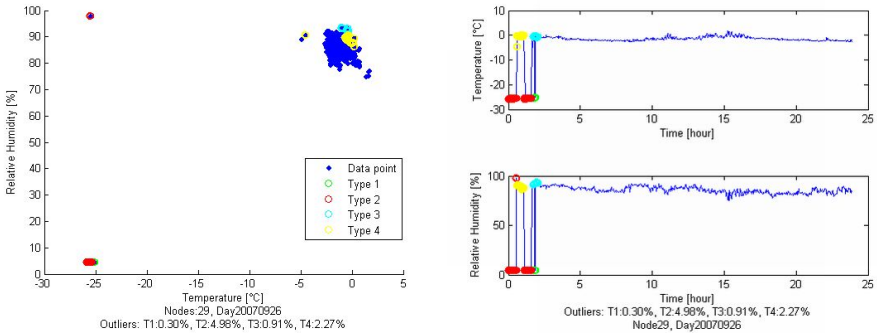


Figure 3.36: Performance of running average-based labelling technique for extreme values

3.5.2 Complexity Comparison

Complexity is another factor to be taken into account while selecting a labelling technique. The complexity of the Mahalanobis distance-based labelling technique is $O(nd^2)$, where n is the number of samples and d is the number of dimensions. The complexity of both density-based and running average-based labelling techniques is $O(n)$, where n is the number of samples.

3.6 Guideline on Choosing Labelling Techniques for Datashapes

Based on our extensive experimental results, we can conclude that the types of outliers that are identified by the three labelling techniques, depend on the shape of the dataset. In general, the Mahalanobis distance-based labelling technique is good for detecting outliers of Type 1, and outliers of Type 3 and 4 outside a solid dataset, when there are no extreme values. The density-based labelling technique is very useful for detecting outliers of Type 1 and 3. The running average-based labelling technique does not target a specific type of outliers. It can detect all four types of outliers, but it will not find all the observations that intuitively can be identified as outliers. Table 3.7 summarizes the performance and applicability of different labelling techniques in terms of the type and placement of outliers.

The overlap between these three labelling techniques, as shown in Figure 3.37, is very small. Each circle in Figure 3.37 represents the mean percentage of detected outliers for all the labelled data with two features. Where the circles

Types of Outliers	Specific Targets	Mahalanobis Distance	Density	Running Average
Type 1		+	++	++
Type 2	single short small sequence	+	+	0
Type 2	long sequence/multiple sequences	-	-	0
Type 3	outside single solid cluster	++	++	+
Type 3	inside single solid cluster	- -	- -	+
Type 3	outside single not dense cluster	++	+	0
Type 3	inside single not dense cluster	- -	-	-
Type 3	outside multiple clusters	++	++	+
Type 3	between multiple clusters	- -	++	+
Type 3	between gaps of the edge of a cluster	- -	++	+
Type 4	outside single solid cluster	+	+	+
Type 4	inside single solid cluster	–	–	+
Type 4	outside single not dense cluster	0	-	0
Type 4	inside single not dense cluster	–	0	+
Type 4	outside multiple clusters	0	-	0
Type 4	between multiple clusters	- -	0	+
Type 4	between gaps of the edge of a cluster	- -	+	+

Table 3.7: Comparison of labelling techniques based on types of outliers they detect

overlap, the overlapping techniques agree on a percentage of observations being outliers. The very small overlap between these three labelling techniques confirms our argument in Section 3.4 stating that different definitions for outliers used by the labelling techniques greatly influence the performance of outlier detection techniques.

3.7 Chapter Summary

Based on our experiments, we can conclude that the choice of labelling techniques is very important and has great impact on performance evaluation of outlier detection techniques. Before an outlier detection technique is evaluated, one should characterize both outlier detection and labelling techniques based on the type of outliers it can (or cannot) detect. Also, it is important to know the data that will be searched for outliers. To prevent our outlier detection techniques presented in the later chapters to be very specific for a certain type of dataset

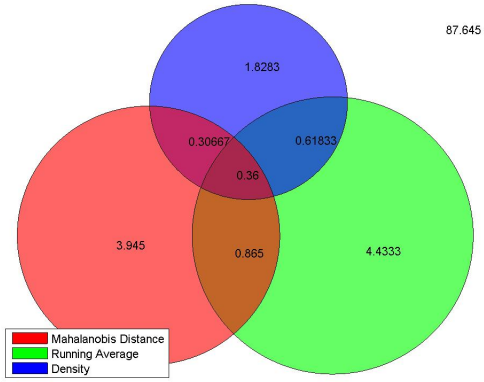


Figure 3.37: Outlier detection rate of the three labelling techniques

and outlier type, we present performance evaluation of our techniques on data labelled by all the three labelling techniques. Having said this, one should keep in mind that for the Grand St. Bernard dataset, in general, the Mahalanobis distance-based labelling technique is good for detecting outliers of Type 1, and outliers of Type 3 and 4 outside a solid dataset, when there are no extreme values. The density-based labelling technique is very useful for detecting outliers of Type 1 and 3. The running average-based labelling technique does not target a specific type of outliers and can detect all four types of outliers, but it will not find all the observations that intuitively can be identified as outliers.

Chapter 4

Statistical-Based Outlier Detection Techniques for Wireless Sensor Networks

Outlier detection techniques are purposefully designed depending on specific applications and corresponding data characteristics. Sensor data collected from densely deployed sensor nodes in the physical environment tends to be correlated in space and time. In this chapter, we efficiently quantify spatial and temporal correlations existing in univariate sensor data and exploit them to propose our statistical-based distributed and online outlier detection techniques for WSNs. Experimental results reveal that taking into account spatio-temporal correlations while designing outlier detection techniques contributes to thorough understanding of the characteristics of sensor data, precise identification of outliers, and detection of changes in normal behavior of sensor data.

4.1 Introduction

As stated in Chapter 2, outlier detection techniques are essentially designed for specific applications and for certain data types. No outlier detection technique can be effectively and efficiently applied to all application domains or datasets [47]. This implies that special characteristics of sensor data should be taken into account while designing outlier detection techniques for WSNs. One of such characteristics is both *spatial* and *temporal* correlation that is hidden in the sensor data [125]. This is especially true in case of environmental monitoring applications [125]. In these applications, in which a large number of sensor nodes are densely deployed in a large area, sensor values collected from geographically adjacent nodes tend to present great resemblance. This feature is called *spatial correlation*. On the other hand, sensor values collected by an individual node tend to present similarity within a time interval. This feature is called *temporal correlation*. We will show that taking into account both these correlations is quite important to design a suitable outlier detection technique for WSN applications. Precise capturing of these correlations helps thorough understanding of normal behavior and specific characteristics of sensor data. Use of the captured correlation in design phase of the outlier detection techniques for WSNs can improve the detection rate and lower down the false alarm. Another advantage of identifying spatial and temporal correlations in the sensor data is introducing the ability to distinguish between different types of outliers, i.e., events and errors.

In this chapter, we investigate spatial and temporal correlations of univariate sensor data and exploit them to propose our statistical-based outlier detection techniques for WSNs. Specifically, we utilize *time series analysis* and *geostatistical data analysis* to obtain temporal and spatial correlations. Our proposed techniques are designed in such a way to operate in distributed manner across multiple sensor nodes with low computational, communication, and memory complexity and to handle continuously collected sensor data in real-time. Experimental results on real environmental dataset collected at the Grand St. Bernard show detection accuracy of our proposed outlier detection techniques as well as their ability to detect changes in normal behavior of sensor data based on the three labelling techniques described in Chapter 3.

The rest of this chapter is organized as follows. Related work on use of spatial and temporal correlations in WSNs is described in Section 4.2. The fundamental methods to quantify temporal and spatial correlations are addressed in Section 4.3. Problems of directly applying traditional spatial and temporal correlation modeling to the WSN and our solutions to that are presented in Section 4.4. Our proposed statistical-based outlier detection techniques are presented in Section 4.5. Experimental results and performance evaluations of our techniques are

reported in Section 4.6. Finally this chapter is concluded in Section 4.7.

4.2 Related Work

Spatial and temporal modelling originate from the field of *statistics*. While both spatial and temporal modelling date back to centuries ago, until recently there has not been a theory about spatial-temporal models except the already well-established theories of spatial statistics and time series analysis [79]. Spatial and temporal models have also been used by *data mining* community. The related work in data mining community can be found in [76, 112, 133, 134]. The purpose of spatial and temporal data mining methods is to extract implicit knowledge and patterns stored in spatio-temporal databases. Nevertheless, the main problem of majority of existing techniques is that they either consider spatial correlation in the dataset but ignore temporal dependency of data, or consider temporal correlation in the dataset without examining spatial relationship among data points.

Recently, researchers have realized the significance of taking advantage of spatial and temporal correlations existing in sensor data for WSN applications, e.g, geographical monitoring, fire detection, target detection and tracking. Vuran et al. [125] exploit spatial and temporal correlations of field source for the realization of efficient medium access and reliable event transport protocols for WSNs. Vuran and Akan [126] further extend this work in [125] by considering the joint effects of spatio-temporal correlation for both point and field sources. Solis and Obraczka [113] use spatial and temporal correlations to introduce an energy-efficient data collection technique for WSNs.

Admittedly, there are also few work on using spatio-temporal correlation for outlier detection in WSNs. Elnahrawy and Nah [28] and Ni and Pottie [82] propose Bayesian-based space-time approaches for fault detection in WSNs. However, they did not explicitly calculated and used the spatial and temporal correlations in a quantitative way and only made the assumption that such correlations exists. Their fault detection techniques also pay very little attention to the characteristics of dynamic and distributed nature of sensor data streams.

Most, if not all, of existing related work have made a similar assumption without further exploration of such correlation. This is due to the fact that capturing the dynamic spatio-temporal correlations of sensor data has high computation, communication, and memory overhead, which sensor nodes cannot afford. To the best of our knowledge, our work is the first attempt to efficiently capture both spatial and temporal correlations using geospatial statistics and time series analysis in a distributed and online manner and have it as integrated part of

distributed and online outlier detection techniques for WSNs.

4.3 Principles of Modelling Spatial and Temporal Correlations

In this section, we describe how to spatially and temporally model the sensor data. These models will be used later in our proposed statistical-based outlier detection techniques in Section 4.5. But first, Subsection 4.3.1 and 4.3.2 briefly describe time series analysis and geospatial statistics for modelling temporal and spatial correlations, respectively.

4.3.1 Modelling Temporal Correlation

We utilize time series analysis to model temporal correlation. A time series is a sequence of values $X = \{x(t) : t = 1 \dots n\}$ that follow a non-random order and the n consecutive values of a variable are taken at *equally spaced* time intervals [21]. The three main goals of using time series analysis are to: (i) understand trend and seasonal changes over time, (ii) understand and model the temporal correlation structure of data, (iii) predict (forecast) further values using the fitted temporal model [21]. The general process of modelling temporal correlation and predicting has the following steps:

Trend and Seasonality Analysis and Removal

The foremost requirement for starting with time series modeling is to achieve a *stationary* time series. A time series is considered to be stationary if its mean μ, variance σ^2, and its autocorrelation structure remain constant over time [21]. The autocorrelation represents the correlation between any two values of $x(t)$ and $x(t + h)$, where h is a time lag. For this purpose, the original time series needs to be plotted and analyzed to see whether it contains any trend or seasonality. Figure 4.1 (left) illustrates an example of trend and seasonality in a time series. Once any trend or seasonality is detected, they should be removed to obtain a stationary time series. In general, the trend can be eliminated using *polynomial fitting*, *moving averages*, *differencing*, or *double exponential smoothing* [13]. The seasonality, on the other hand, can be eliminated using *linear model fitting* or *differencing* [13]. By applying these techniques to the time series, the stationary time series is successfully achieved. Figure 4.1 (right) illustrates the corresponding stationary time series of Figure 4.1 (left) after trend and seasonality removal. The

residuals in this figure constitute the stationary time series that has a constant mean and variance over time.

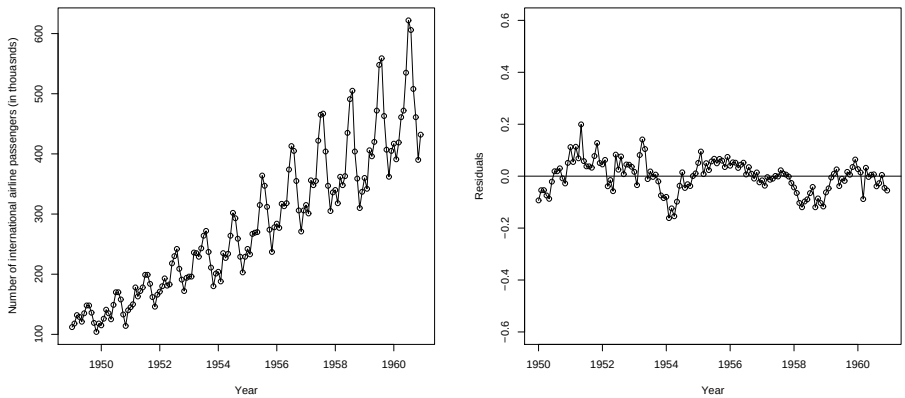


Figure 4.1: (left) Example of trend and seasonality in a time series, (right) Corresponding stationary time series after trend and seasonality removal

Time Series Modelling

Next step is to define a model to fit the stationary time series. A commonly used model for doing so is *autoregressive moving average* (ARMA), also known as *Box-Jenkins approach* [21]. The ARMA model is an efficient tool for modeling and forecasting a time series [13]. ARMA model consists of an *autogressive* component (AR) and a *moving average* component (MA). AR component is simply a linear regression of the current value *x(t)* against its p prior values in the time series, where p is the order of the autoregressive component. MA component, on the other hand, is simply a linear regression of the current value *x(t)* against the white noise of its q prior values in the time series, where q is the order the moving average component. The *white noise* is defined as a sequence of uncorrelated random variables having $N(0,1)$ distribution. The complete ARMA(p,q) model is formulated as:

$$x(t) = \alpha_1 x(t-1) + \ldots + \alpha_p x(t-p) + z(t) + \beta_1 z(t-1) + \ldots + \beta_q z(t-q) \quad (4.1)$$

where $\{z(t-q), \ldots, z(t)\}$ is a white noise sequence, and $\alpha_i = \{\alpha_i : i = 1 \ldots p\}$ and $\beta_j = \{\beta_j : j = 1 \ldots q\}$ are a sequence of constant parameters for AR(p) and MA(q) models, respectively. Appropriate values for p and q can be defined either manually by observing the plot of the autocorrelation structure and partial autocorrelation structure [13] or automatically using a special software. After specifying p and q, parameters α_i and β_j can be calculated using *non-linear least squares* or *maximum likelihood estimation* [21]. To check whether the defined ARMA correctly fits the stationary time series, its residuals should be analyzed to have a normal distribution.

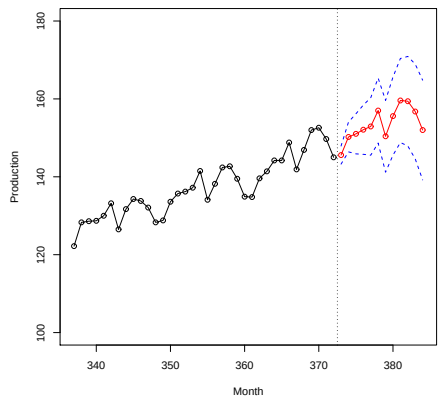


Figure 4.2: Prediction for future 12 months. The vertical dotted line separates the data from the predictions.

Time Series Prediction

Once a correct fitted ARMA model is defined, future values of the time series can be predicated using the model, as well as previous values, and the white noises, based on Equation 4.1. A future value predicted by the ARMA model is often within a *confidence interval*, which is used to indicate the reliability of the predicted value [21]. The confidence interval usually uses the *standard error* of the predicted value to represent its *confidence level*. The standard error is the square root of the estimated standard deviation [13]. One standard error gives 66.7% confidence level. Two standard errors gives 95% confidence level, which means that the probability that the true value lies in the confidence interval of its

predicted value is 95%. Three standard errors gives 99.7% confidence level. Using the ARMA model for prediction, a predicted value will be obtained as well as its upper bound and lower bound based on the given confidence level. Figure 4.2 illustrates the predicted values and their corresponding confidence intervals at future time periods.

4.3.2 Modelling Spatial Correlation

To model spatial correlation, we use geostatistical data analysis [22]. *Geostatistical data* is a sequence of values $Y = \{y(s) : s = 1 \dots m\}$ collected from m *coordinate-aware* locations at a time interval [22]. The three main goals of using geostatistical data analysis are to: (i) understand trends in spatial data, (ii) understand and model the spatial correlation structure of data, (iii) predict the data value at a location nearby [22]. The general process of modelling spatial correlation and prediction has the following steps:

Trend Analysis

Spatial data may present two types of spatial structures, i.e., trends and local spatial dependency. Trends present themselves in geostatistical data collected from all locations of monitored geographical space. Trends may have different forms, i.e., *the first-order trend* (like a plane), *the second-order trend* (like a bowl or a dome), or *the higher-order trend*. These trends can be visually depicted by the *postplot* [135], which shows the relationship between the relative values or colors (or both) and the locations of each observation. Figure 4.3 illustrates the first-order trend surface (linear), where the size of the relative values (related to the attribute values) as well as the color systematically change over the entire geographic space. Once the trend is detected, a linear regression can be used to compute the trend and to define the spatial model for the entire region [135].

Local Spatial Dependency Modelling

The second type of spatial structure is the *local spatial dependency*, which represents the similarity between observations collected at adjacent locations in a local region. The most common tool to model the local spatial dependency is the *variogram* [22], which represents the relationship between the *separation distances* and the corresponding *semivariances* of two observations. The separation distance is the geographical difference between any pair of locations (s_i, s_j). The semivairance indicates the mathematical difference of any pair of observations $(y(s_i), y(s_j))$, which is calculated by half the squared difference between both

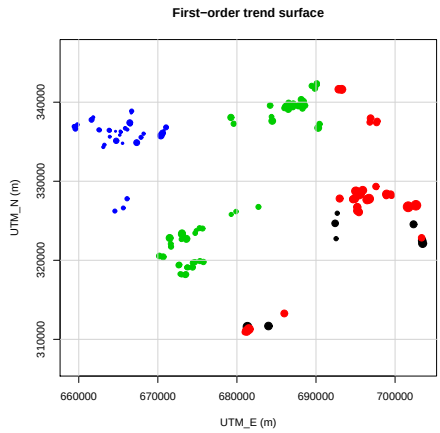


Figure 4.3: Example of the first-order trend surface

values. In a variogram, the closer two locations are, the smaller the semivariance between their observations. There are two kinds of variograms, i.e., the *variogram cloud* and the *empirical variogram* (*semivariogram*) [22]. Figure 4.4 (a, b) illustrates both variograms, from which it can be seen that the variogram cloud displays the relationship between the separation distances and the semivariances of all pairs of observations. The empirical variogram, on the other hands, depicts the relationship between the grouped separation distances in the forms of *bins* and the average semivariances of all pairs of observations in these bins. In contrary to the variogram cloud, the empirical variogram better and more clearly illustrates the local spatial dependency [135].

The empirical variogram has three unique parameters, that are, the *sill*, the *range* and the *nugget*. These three parameters can characterize a local spatial dependency. The sill is the maximum semivariance value, which represents the values at the locations with no spatial dependency. The range is the corresponding separation distance at which the sill is reached. It represents the maximum distance at which there is no evidence of the spatial dependency. The nugget is the semivariance value as the separation distance approaches zero. When the sill is equal to the nugget in the empirical variogram, it indicates there is no local spatial dependency. This state is called a *pure nugget effect* [114].

If the empirical variogram represents a local spatial dependency, a *theoretical variogram* model should be defined to fit this variogram. For doing so, there exists various models such as *spherical*, *circular*, *exponential*, *linear*, or *gaussian*

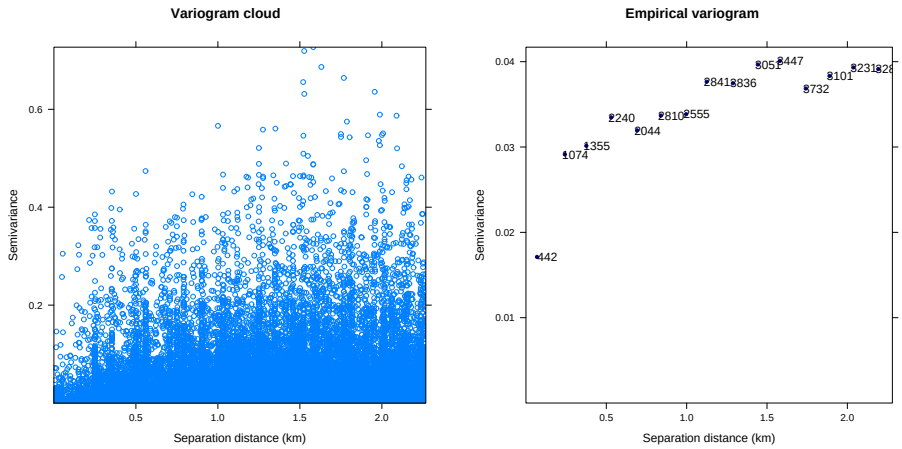


Figure 4.4: (left) Example of a variogram cloud, (right) Corresponding empirical variogram

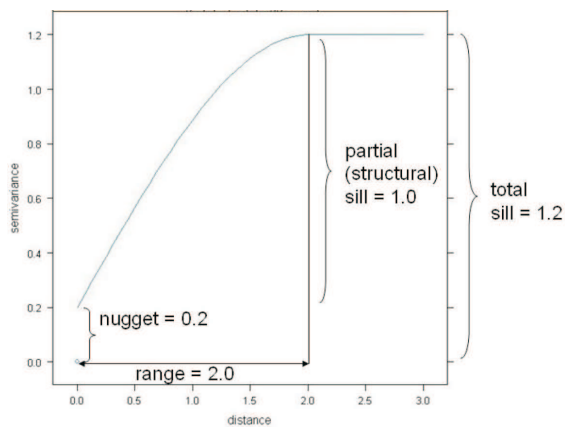


Figure 4.5: Example of a spherical variogram model

models [135]. These models are parameterized by the sill, the range and the nugget. Figure 4.5 illustrates a spherical variogram model fitting an empirical variogram. After selecting one of theoretical variogram models, its parameters can be estimated by either *non-linear least squares* or *maximum likelihood estimation* [114].

Geostatistical Data Prediction

Once the selected theoretical variogram has been precisely estimated, an unknown observation $y(s_0)$ at a known location s_0 can be predicted using this fitted variogram as well as coordinates of neighboring locations and their observations. This process is known as *kriging* [114], which is an optimal geostatistical interpolation technique and provides the best linear unbiased predictor (BLUP) for every location. By using kriging, the unknown observation is predicted as a linear weighted combination of observations collected at its adjacent locations. Its predicted value $\hat{y}(s_0)$ is formulated as:

$$\hat{y}(s_0) = \lambda_1 y(s_1) + \ldots + \lambda_m y(s_m) \tag{4.2}$$

where $\{y(s_1) \ldots y(s_m)\}$ are the observations at the adjacent locations of s_0 and $\{\lambda_1 \ldots \lambda_m\}$ are the *weights* in terms of semivariance values between s_0 and its adjacent locations, $\sum_{i=1}^{m} \lambda_i = 1$. Their semivariance values corresponding to the distance between them can be obtained from the estimated variogram. A detailed description of weights calculation can be found in [116]. The predicted value $\hat{y}(s_0)$ has a confidence interval based on a given confidence level. The standard error of $\hat{y}(s_0)$ is used to represent the confidence level.

4.4 Fitting Spatial and Temporal Correlations Modelling to Resource-Constraint WSNs

It is obvious that the entire process of spatial and temporal correlations modelling described in Section 4.3 has high computational and memory complexity as well as high communication overhead. Such high complexity and overhead is not affordable by the strongly resource-constraint sensor nodes and WSNs. Furthermore, choosing various parameters as appropriately as possible requires interaction with experts. These are not in line with the requirements of distributed data processing and self-organizing operation of WSNs. Therefore, to use this intricate and time-consuming process in WSNs, it needs to be highly simplified in terms of resource consumptions and automation. After all, the precision of modelling spatial and temporal correlations themselves is not our most important focus but a

means to enhance our outlier detection mechanisms. Furthermore, fitting "best" model to historical data may not give best forecasting values [21].

In what follows we present practical problems while modelling both spatial and temporal correlations in WSNs and give our solutions to alleviate them and to make the modelling process local and distributed. We consider a relatively small sub-network consisting of densely deployed n sensor nodes $\{s_1, \ldots, s_n\}$, in which observations are made at (nearly) equal time intervals, all nodes can directly communicate with each other by the radio transmission, and each node knows its own location as well as location of its *n-1* adjacent neighbors. Figure 4.6 illustrates an example of such sub-network. The reason of making such assumption is that this sub-network can easily be extended to other types of network topologies. For instance, a clustering-based network topology can be obtained by inserting a cluster-head for controlling this sub-network, or a hierarchal-based network topology can be made by adding a parent node for taking care of its children nodes being located in this sub-network. The Grand St. Bernard deployment we described in Chapter 3 has a such small cluster, which consists of 5 densely deployed sensor nodes with a base station.

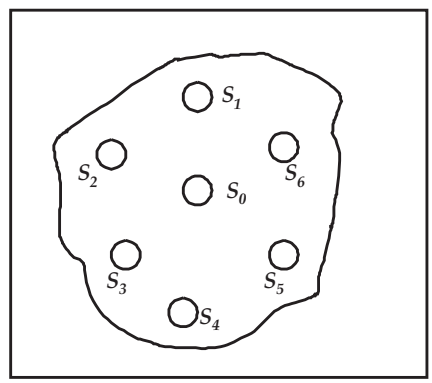


Figure 4.6: Example of a sensor sub-network

We use $x(s,t)$ notation to present location information (s) of each node in the sub-network and the time epoch (t) at which the observation is made.

4.4.1 Modelling Temporal Correlation in WSNs

It is quite likely that a time series contains some errors or missing values, which is a very probable case for sensor data. Using such imprecise time series will

result to an unreliable AR model and incorrect future value prediction. To solve this problem, we first use *smoothing* techniques on the original time series before modelling temporal correlation. This kind of techniques can effectively correct errors and give reliable values for missing data. A commonly used smoothing technique is *median smoothing* [14], which replaces errors and missing data using median values within each smoothing window. The choice of the size of smoothing window will be discussed in our experiments.

The precondition of modeling temporal correlation is to obtain a stationary time series, where any trend or seasonality is removed. One of efficient non-parameteric techniques can be used to eliminate the trend. Due to its low computational complexity, we use *first differencing*, which defines a new time series as $X^{'} = \{x^{'}(s,t) = x(s,t) - x(s,t-1) : t = 2 \dots n\}$ [21].

Analyzing seasonality is a rather complicated task, which requires a large quantity of data collected in a considerably long period of time (in the order of several years or decades). Even if this amount of data is available, the process of analyzing seasonality also needs much knowledge from experts. The Grand St. Bernard dataset we discuss in this chapter is collected in approximately two months, which implies that there is no significant seasonality. Assuming having no seasonality lowers down a great deal of computational and memory complexity to analyze and remove seasonality. However, in fact, sensor observations of a variable collected in each day has somewhat a *repetitive* pattern. For instance, temperature usually attains its minimal value in a day around the sunrise and reaches its maximal value on the same day around 2pm. The Grand St. Bernard dataset also follows this principle. Figure 4.7 illustrates the distinct upward and downward trends occurring at different time periods during one day. It, moreover, shows that the trend of the previous day is apparent at the corresponding time periods of the next day. This motivates our assumption of using the model of temporal correlation of each day to predict the values of the next day. One should note that using the temporal correlation of the whole day to predict values of the entire next day would lead to different results than using models of each time period to predict the values of the next day at the corresponding time period. Thus, for instance, temporal correlation modelling of the period of 6am-14am (an upper trend) of one day can be used to predict the values of the same time period on the next day. This time-wise modelling and prediction improves reliability of value prediction and enhances the outlier detection process.

The significant complexity of ARMA model stems from the fact that it uses as many as possible parameters to precisely define the temporal model with minimum estimation errors. To simplify the ARMA, we only use the AR model, which implies that the current observation is only correlated to its previous p values (and not to the white noise represented by q). Consequently, Equation 4.1

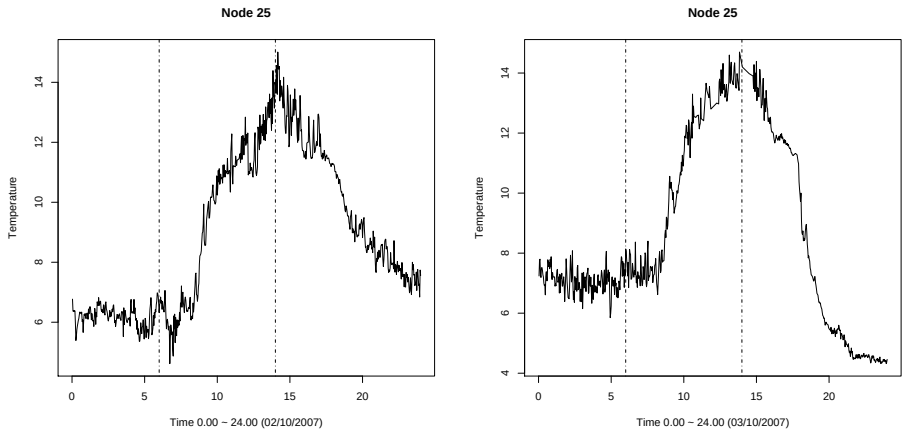


Figure 4.7: Example of a repetitive pattern occurred in two consecutive days in the Grand St. Bernard dataset

is simplified to:

$$\hat{x}(t) = \alpha_1 x^{'}(t-1) + \ldots + \alpha_p x^{'}(t-p) \tag{4.3}$$

This AR model has often been used [21]. To decrease the complexity of estimating the constant parameters $\{\alpha_i : i = 1 \ldots p\}$ as well as ensure the reliability of the estimated model, we specify a relatively small p. We will investigate impact of different p values on performance of outlier detection techniques in the experiments. In this way, the process of modelling temporal correlation is simplified to an extent that each individual sensor node can carry it out locally. Each node exploits the estimated AR model and its previous observations to predict the next new observation.

The confidence level of a predicted value depends on the standard error. To ensure the reliability of the predicted data despite of using a simplified model, we use a relatively large confidence level for predicted value, e.g., 95% confidence level (two standard errors), being denoted as $[\hat{x}(s,t) - 2\hat{\sigma}, \hat{x}(s,t) + 2\hat{\sigma}]$. In other words, each true observation will fall in the confidence interval of its corresponding predicted values with the probability of 95%. Impact of different confidence levels on performance of outlier detection techniques will be investigated in our experiments.

4.4.2 Modelling Spatial Correlation in WSNs

Compared to temporal correlation modelling, spatial correlation modelling in an efficient manner is a more challenging task for the WSNs. Analyzing a possible trend in an entire region requires massive data communication and transmission from all the sensor nodes. Even if we assume that all this data can be transmitted and processed locally in a central place, finding a trend for the entire region may not be easy [135]. By extensive data analysis, we have stated in Chapter 3 that there is obviously no trend existing in the Grand St. Bernard dataset. Here spatial exploratory plots at different time instants also indicate that no trend occurs in the Grand St. Bernard dataset, as shown in Figure 4.8.

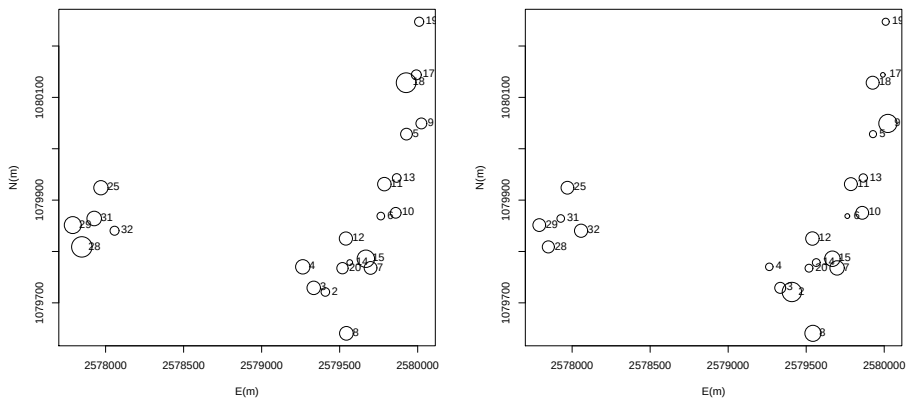


Figure 4.8: Spatial exploratory plots at different time instants in the Grand St. Bernard dataset. The size of circle corresponds to the value of each observation.

As it was mentioned before, a local spatial dependency can be modelled by the empirical variogram. Building an empirical variogram in WSNs is a serious challenge, since the reliability of a generated empirical variogram depends on the number of observation pairs [135]. In general, at least 100∼150 pairs of observations are required to reliably produce an empirical variogram [135]. This centralized analysis of a large amount of data causes too much communication overhead and is not suitable for WSNs. Furthermore, lack of sufficient observation pairs at each time epoch causes the failure of building a reliable empirical variogram. To alleviate this problem, we take the method used in [114], which combines more observations at different time periods from the limited number of

locations to constitute the empirical variogram. This method is justified based on the assumption that observations collected at different time periods can be characterized by a spatial correlation structure, which is modelled by the same empirical variogram [114]. In this way, each pair of observations collected at the same time period contributes to a combined empirical variogram. Combined semivariance can be formulated as:

$$\hat{\gamma}(h) = \frac{1}{2n(h)} \sum_{t=1}^{m} \sum_{s=1}^{n_t(h)} (y(s,t) - y(s+h,t))^2 \tag{4.4}$$

where m is the number of different time periods, h is a distance lag, and $n_t(h)$ denotes the number of point pairs for each h at time period t.

Afterwards, a theoretical model should fit the empirical variogram. Likewise, finding a correct fitted model for the variogram is a quite time-consuming task and may not be best for future values prediction. Thus, we use a simple theoretical model, e.g., linear model, to fit the variogram and to reduce the computational complexity of capturing the semivariances for prediction. The bounded linear variogram model can be defined as

$$r(h) = \begin{cases} c(h/a) : h < a \\ c : h \geq a \end{cases} \tag{4.5}$$

where c is the sill parameter, a is the range parameter of the bounded linear model and h is a distance lag.

As stated before, predicting values of an unsampled location requires the weights of its locally spatial neighbors as well as their observations. One should note that for each node, the weights of its corresponding neighbors need to be calculated. This implies that calculating weights for n nodes is relatively expensive. To solve this problem, a gateway node equipped with much more computational power and memory storage can be used. The base station deployed in the small cluster of the Grand St. Bernard can play this role to collect spatial data and calculate weights for nodes. Otherwise, one of sensor nodes having more energy and memory can be used. Once the weights of neighbors of each node are calculated and delivered, each node can predict its own value using these weights and their observations. Similarly to temporal correlation modelling, we use a relatively large confidence level, e.g., 95% confidence level (two standard errors) for predicted values in the spatial correlation model, which is denoted as $[\hat{x}(s,t) - 2\hat{\sigma}, \hat{x}(s,t) + 2\hat{\sigma}]$.

4.5 Statistical-Based Outlier Detection Techniques

In Section 4.4, we presented solutions to efficiently model spatial and temporal correlations of sensor data to meet special requirements of WSNs. In this section, we present our distributed and online statistical-based outlier detection techniques for WSNs by taking advantage of temporal correlation, spatial correlation and spatio-temporal correlation, respectively. We will show high accuracy of our outlier detection techniques in terms of identifying outliers and detecting changes in normal behavior of the sensor data as well as their ability to update the spatial and temporal models on the fly.

The main goal of our outlier detection techniques is to enable each node to utilize predicted values estimated by the defined spatio-temporal correlation models to classify their new sensor observations as either outlier or normal in an online manner and detect any changes in normal behavior of the sensor data upon occurrence. In what follows, we also provide some insight on how to distinguish between various types of outliers as defined in Chapter 3.

4.5.1 Temporal Correlation-Based Outlier Detection Technique (TOD)

We here present our temporal correlation-based outlier detection technique (TOD), which instantly identifies temporal outliers in a single node using temporal correlation of local data. A *temporal outlier* is an actual observation $x(s,t)$ exceeding the confidence interval of its predicted value $\hat{x}(s,t)$ with a given probability.

Upon completion of temporal correlation modelling and value prediction, it is now time to identify outliers. To clarify TOD, we consider three questions, that are, (i) how to deal with $x(s,t)$ after identifying it as normal or outlier. (ii) how to distinct $x(s,t)$ as an error or indication of the occurrence of an event, if $x(s,t)$ is identified as an outlier. (iii) when and how to update the temporal model to better capture changes in normal behavior of sensor data.

For the first question, identified normal observations can be directly used for the next prediction. Otherwise, after detecting an outlier $x(s,t)$, TOD uses observations at the previous time instances $\{x(s,t-p),\ldots,x(s,t-1)\}$ and the temporal correlation model to identify the next observation sequence as normal or outlier. The standard error of predicted values will be incrementally updated step by step. Here two possibilities exist:

- If all observations of the entire sequence are identified as outliers, the entire sequence represents occurrence of an event and indicates changes in normal

behavior of sensor data. Consequently, this observation sequence including $x(s,t)$ will replace $\{x(s,t-p),\ldots,x(s,t-1)\}$ for the future prediction.

- If only a few observations in the sequence are detected as outliers, we consider the observations labelled as outlier as errors and will not use them for the next prediction process. Instead, the predicted values of these errors are used to predict the next observation. In case of identifying absolute errors, the predicted values will directly replace the outliers in the next prediction process.

Being able to correctly label the sequence as event or error is very important because an error-labelled sequence will be replaced by a new sequence of predicted values, while an event-labelled sequence will be directly used in the following prediction process.

We further discuss how to decide about the length of the observation sequence to distinguish between errors or events. This actually is a practical problem depending on application requirements and sampling rates. One may note that the duration of an event is not known beforehand. Moreover, some events last longer than the others. These characteristics make identification of the entire event difficult. Therefore, we aim at identifying changes in the normal behavior of data instead of identifying the entire event. This implies that the length of the observation sequence is not necessary to be long. As stated in Chapter 3, we chose the minimum length of a sequence to be four observations to distinguish between different types of outliers in the Grand St. Bernard dataset. This implies that each observation sequence lasts for 8 minutes.

TOD waiting to distinguish between events and errors does not cause a considerable delay in the process of identifying type of outlier, as each new observation is labelled as outlier or normal upon arrival. Spending a short delay for making this distinction is unavoidable. In this way, the second question is completely answered.

For the third question regarding when and how to update the temporal model, changing this parametric temporal model requires the same amount of observations to perform the model update. Updating the model consumes high resources in terms of memory and processor. To solve these problems, we can update the model only at the end of the day. After outlier detection is finished for one day, we have a new clean time series consisting of the original normal observations, the event-based observations, and the predicted values which replace the detected errors. This new time series can be directly used to model the possibly changed time correlation without any smoothing operation. Each node can check whether any event has been detected in the whole time series. Otherwise, the existing temporal correlation model can still be used for the following day. In this condition,

```
1 procedure FittingTemporalModel(time series data)
2   each node smoothes time series data;
3   each node models temporal correlation by fitting a AR model;
4   initiate OutlierDetectionProcess(AR model parameters);
5   return;
6 procedure OutlierDetectionProcess(AR model parameters)
7   when a new observation x(s,t) arrives
8       predict x̂(s,t) using AR model parameters as well as previous observations;
9       if(x(s,t) > (x̂(s,t) + ci * σ) or x(s,t) < (x̂(s,t) − ci * σ))
10          x(s,t) indicates a temporal outlier;
11          successive outliers + 1;
12          use the same previous observations for the next prediction;
13          if(the number of successive outliers > the length of time sequence)
14              it indicates a change in normal behavior of sensor data;
15              these successive outliers are used for the next prediction;
16          endif;
17      else
18          if(the number of successive outliers > 0)
19              predicted values of previous temporal outliers are used for the next prediction;
20          endif
20          x(s,t) indicates a normal observation and is used for the next prediction;
21      endif;
22      if(at the end of the day)
23          initiate UpdatingTemporalModel(time series data);
24      endif
25  return;
26 procedure UpdatingTemporalModel(time series data)
27  refit a temporal correlation AR model;
28  return;
```

Table 4.1: Pseudocode of TOD

each node keeps on using the model until a significant change in the behavior of data is detected. This can contribute to reduction of number of model updates. The corresponding pseudocode for TOD is shown in Table 4.1.

TOD enables that each node analyzes its local data to detect outliers in an online manner and to further distinguish between different types of outliers in a timely manner. Although its detection performance may be impacted by having only local data analysis, its local processing and low resource consumption makes it ideal for WSNs. An offline outlier detection requires existence of the entire time series and has a considerable detection delay. It also fails in detecting changes in behavior of normal data or between two consecutive time series [93] as soon as it occurs. In contrary, our TOD detects outliers upon arrival of a new observation and can also solve the problem of occurrence of missing values, which can be timely replaced by reliable predicted values.

4.5.2 Spatial Correlation-Based Outlier Detection Techniques (SROD and SPOD)

We here present our two spatial correlation-based outlier detection techniques, which enable each node to identify outliers in an online manner and detect the change of normal behavior by only taking advantage of the spatial correlation. As stated before, each node obtains the weights $\{\lambda_m : m = 1 \dots n-1\}$ of its $n-1$ spatial neighbors after modelling the spatial correlation in the local region. Weights indicate the spatial correlation between observations of the neighboring nodes. Then each node uses its spatial neighbors' observations collected at the current time instant together with these corresponding weights to predict its own current value. A *spatial outlier* is an actual observation $x(s,t)$ at this time lies outside of the confidence interval of its predicted value.

The simplest way to obtain neighbors' observations at each time instant is that each node transmits its own observation to all its neighbors at each time instant. We call it as SROD, which uses real observations from spatial neighbors for prediction and spatial outlier detection. However, this frequent data transmission causes a large volume of communication overhead and bandwidth occupation. Moreover, sending and receiving information of all the nodes to all their neighbors lead to a considerable detection delay. This makes identification of outliers a rather time-consuming task. In addition, this continuous data transmission may bring a potential negative impact for operation of other protocols in WSNs. Considering these disadvantages, we propose the other strategy (SPOD) to predict spatial neighbors' observations without any actual observations transmission.

To do so, at the beginning of spatial outlier detection process, each node transmits its own parameters $\{\alpha_i : i = 1 \dots p\}$ indicating its temporal correlation

model to its spatial neighbors. Once each node receives these parameters from its neighbors, it first uses them together with its own previous observations (based on Equation 4.3) to predict the current values for its neighbors. Afterwards, each node uses the newly predicted value of each of its neighbors together with their corresponding weights (based on Equation 4.2) to predict its own current value. This value is a linear combination of predicted values for its neighbors along with their weights. Accordingly, those actual observations of each node exceeding the confidence interval of the predicted value are considered as outliers. In this way, our SPOD is performed at each node without any actual observation transmission and only by transmitting a few parameters. The computation load of SPOD is also low because it only involves prediction, which is a linear operation.

Once a node identifies the entire observation sequence as outliers, it sends a notification message about the occurrence of an event to all its neighbors. Upon receipt of a positive confirmation about occurrence of this event from its neighbors, it confirms occurrence of an event. Otherwise, it considers the observation sequence as long-term errors and uses the predicted values to replace them in the next prediction process.

We assume that location of our sensor nodes is fixed. This implies that the relationship between semivariance values and location is relatively stable. Therefore, the estimated model based on semivarience and location as well as the derived spatial correlation between nodes (weights) can require no update. Each node keeps the weights indicating the spatial correlation for spatial outlier detection.

SPOD only requires a limited transmission of the parameters without transmitting any actual observations. This lowers down the communication overhead for the WSNs. The fact that the spatial model is not updated helps in reducing the computation and memory overheads. Both SROD and SPOD also enable each node to detect outliers and further distinguish between different types of outliers including long-term errors.

4.5.3 Spatio-Temporal Correlations-Based Outlier Detection Techniques (TSOD and STGOD)

We have separately introduced our temporal correlation-based and spatial correlation-based outlier detection techniques. Here we present our two spatio-temporal correlations-based outlier detection techniques. The straightforward way to identify spatio-temporal outliers is that each node locally identifies temporal outliers and then check whether these temporal outliers are spatial outliers by obtaining neighbors'observations at corresponding time instants of temporal outliers. We call it as TSOD, which separately takes advantage of temporal and spatial

correlations of sensor data for outlier detection.

Although our temporal and spatial outlier detection techniques are both online and distributed and are enable to detect changes in the normal behavior of data, they are not quite complete and have their own disadvantages. TOD fully relies on the local data of a single node. Therefore, it does not have enough information to distinguish well between long-term errors and events. Even if it can detect events, they may not be highly reliable due to the fact that the information from neighboring nodes is ignored. In contrary, SROD and SPOD, which are concerned about a local neighborhood, is more reliable on detecting events and can also distinguish them well from long-term errors. SROD obtains neighbors' observations at each time instant and this frequent data transmission causes a large volume of communication overhead and bandwidth occupation. SPOD uses predicted values to replace the actual observations of spatial neighbors to determine spatial outliers. Although this manner avoids actual data transmission during the outlier detection, it still causes the value difference between the actual observations and predicted values for spatial neighbors. Moreover, spatial outlier detection considers spatial neighbors whereas ignoring the temporal correlation of the local node itself, which probably brings a negative impact on the final detection results, especially when some of nodes are suffering from the network inference. Considering these drawbacks of temporal outlier detection and spatial outlier detection, we endeavor to integrate both of them, and make good use of their advantages while reducing the negative impact.

Our STGOD is motivated by the idea that gathering information in a central base station and analyzing them at once allows having more information and better identification of outliers. We aim to bring the same concept into the WSNs by devising a distributed and online spatio-temporal correlations-based outlier detection technique capable of not only making a more reliable identification of global outliers but also distinguish between errors and events. The main steps of STGOD are:

- *Step 1.* Spatial correlation between each node and its neighborhoods will be calculated by geostatistics data analysis and sent to the nodes. As a result, each sensor node will have the corresponding weights of its $n-1$ spatial neighbors $\{\lambda_m : m = 1 \dots n-1\}$.

- *Step 2.* Each sensor node s_j models its temporal correlation using the time series analysis and obtains parameters $\{\alpha_j : j = 1 \dots p\}$ of the temporal model.

- *Step 3.* Each node s_j communicates parameters $\{\alpha_j : j = 1 \dots p\}$ of its temporal model to its spatial neighbors. Thus, each node communicates p

elements over the network.

- *Step 4.* Each node combines the parameters indicating the temporal correlation of its neighbors with their weights indicating the spatial correlation. The integrated parameters are denoted as $\{\sum_{i=1}^{n-1} \alpha_{i1}\lambda_{i1}, \ldots, \sum_{i=1}^{n-1} \alpha_{ip}\lambda_{i(n-1)}\}$, which represents the spatial integration. This integration avoids using traditional median or mean. Being dependent on on the actual spatial correlation, make the integrated parameters more reliable and reasonable.

- *Step 5.* The spatial integration of the parameters are integrated with those of each node s_j itself. The complete integrated parameters are denoted as $\{\frac{\alpha_{j1}+\sum_{i=1}^{n-1} \alpha_{i1}\lambda_{i1}}{2}, \ldots, \frac{\alpha_{jp}+\sum_{i=1}^{n-1} \alpha_{ip}\lambda_{i(n-1)}}{2}\}$, which combines both spatial and temporal correlations.

- *Step 6.* Each node uses its parameters $\{\alpha_g : g = 1 \ldots p\}$ together with its own previous observations to predict its next observation using Equation 4.3. It then compares the predicted value with its actual observation and accordingly identifies the actual observation as global outlier or normal data.

Afterwards, STGOD can identify various of outliers in real-time using the same strategy described in TOD. Also, the temporal correlation model can be updated at the end of the day.

STGOD scales with increase of number of nodes due to its distributed processing nature. Moreover, STGOD enables each node to detect outliers in an online manner with higher reliability. This is achieved by integrating its local temporal correlation with the spatial correlation from its spatial neighbors. STGOD can also distinguish between errors and events in a timely manner as it combines both local and spatial information. This whole process has low communication overhead as well as computational and memory complexity, and does not need to transmit any actual observations between sensor nodes only the parameters.

4.6 Experiments

This section describes performance evaluation of our TOD, SROD, SPOD, TSOD and STGOD techniques. The goals of our evaluation are (i) to test the accuracy of our distributed and online outlier detection techniques and their performance in terms of parameters selection, (ii) to test the detection of changes in normal behavior of sensor data, and (iii) to investigate impact of different labelling techniques described in Chapter 3 on performance of outlier detection techniques.

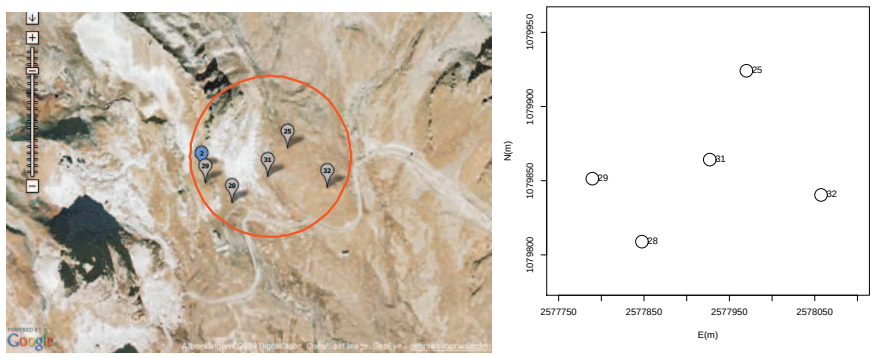


Figure 4.9: The small cluster of the Grand St. Bernard deployment [108] and their corresponding metric coordinates (E-N)

4.6.1 Experimental Dataset

In our experiments, we use the real dataset collected from the small cluster of the Grand St. Bernard deployment for the period of 6am-14am on two consecutive days (29^{th} and 30^{th} September 2007) with one attribute: ambient temperature. Figure 4.9 illustrates the small cluster, which consist of node 25, 28, 29, 31 and 32, and their corresponding metric coordinates. We label the dataset of 30^{th} September (240 observations for each node) using different labelling techniques of Chapter 3, i.e., based on Mahalanobis distance, density and running average for temporal, spatial and spatio-temporal outliers. For temporal labelling, we use the three labelling techniques for each node individually to identify outliers among the dataset of the node for the entire period. For spatial labelling, we use the three labelling techniques for all nodes to identify outliers at each time instant. For the spatio-temporal labelling, however, we use the three labelling techniques on combined dataset of all nodes for the entire period.

We use *R* for the whole simulation. *R* is an open-source language and environment for statistical computing and visualization [101]. The advantages of using *R* for the statistical analysis include (i) it is completely free under the GNU public license and is always available over the internet, (ii) it provides a powerful, programmable, portable, and open computing environment, applicable to the most complex and sophisticated statistical and visualization problems, and (iii) it runs on almost all operating systems (Windows, Unix, Linux and Macintosh) and all source code is published.

4.6.2 Experimental Results and Evaluation

In our experiments, we evaluate two important performance metrics, the detection rate (DR), which represents the percentage of outliers that are correctly detected and the false alarm rate, also known as false positive rate (FPR), which represents the percentage of normal data that are incorrectly considered as outliers. DR represents the ratio between the number of correctly detected outliers and the total number of outliers, while FPR represents the ratio between the number of normal data detected as outliers and the total number of normal data. We present our experimental results and related discussion based on temporal outliers, spatial outliers and spatio-temporal outliers.

Temporal Correlation-Based Outliers

We examine the effect of several important parameters for TOD. These parameters include the size of smoothing window, the order of AR(p) model and the value of confidence level. In our experiments, the size of smoothing window has taken values from $\{15, 30, 48, 60\}$, the order of AR(p) model has varied between from $\{1, 2, 3, 4\}$ and the confidence level has ranged $\{90\%, 95\%, 99\%, 99.7\%\}$. Figure 4.10 illustrates the original time series on 29^{th} September and corresponding smoothed time series using different sizes of smoothing window. We can see that relatively small size of smoothing window can effectively smoothen data as well as keep internal data structure while the large size of smoothing window causes the change of original data structure. Figure 4.11 illustrates the results of applying running average-based labelling technique and temporal outliers (marked as solid circle) detected by TOD for different sizes of smoothing window, where the dashed lines represent the upper bound and lower bound of predicted values. Table 4.2 shows the detection rate and false alarm rate for temporal outliers using three labelling techniques for different sizes of smoothing window. As it can be seen from Table 4.2, larger size of smoothing window results in lower detection rate on all three labelled datasets while false alarm rate is slightly reduced. This is due to the fact that it influences the original data structure and consequently has a negative impact on outlier detection results. To ensure reliable outlier detection results, we set the size of smoothing window to 15 (half of hour) in the experiments.

We also examine the effect of the order of AR(p) model for TOD. Figure 4.12 illustrates the results of applying Mahalanobis distance-based labelling technique and temporal outliers detected by TOD for different orders of AR(p) model. Table 4.3 shows the detection rate and false alarm rate for temporal outliers using three labelling techniques for different orders of AR(p) model. As it can

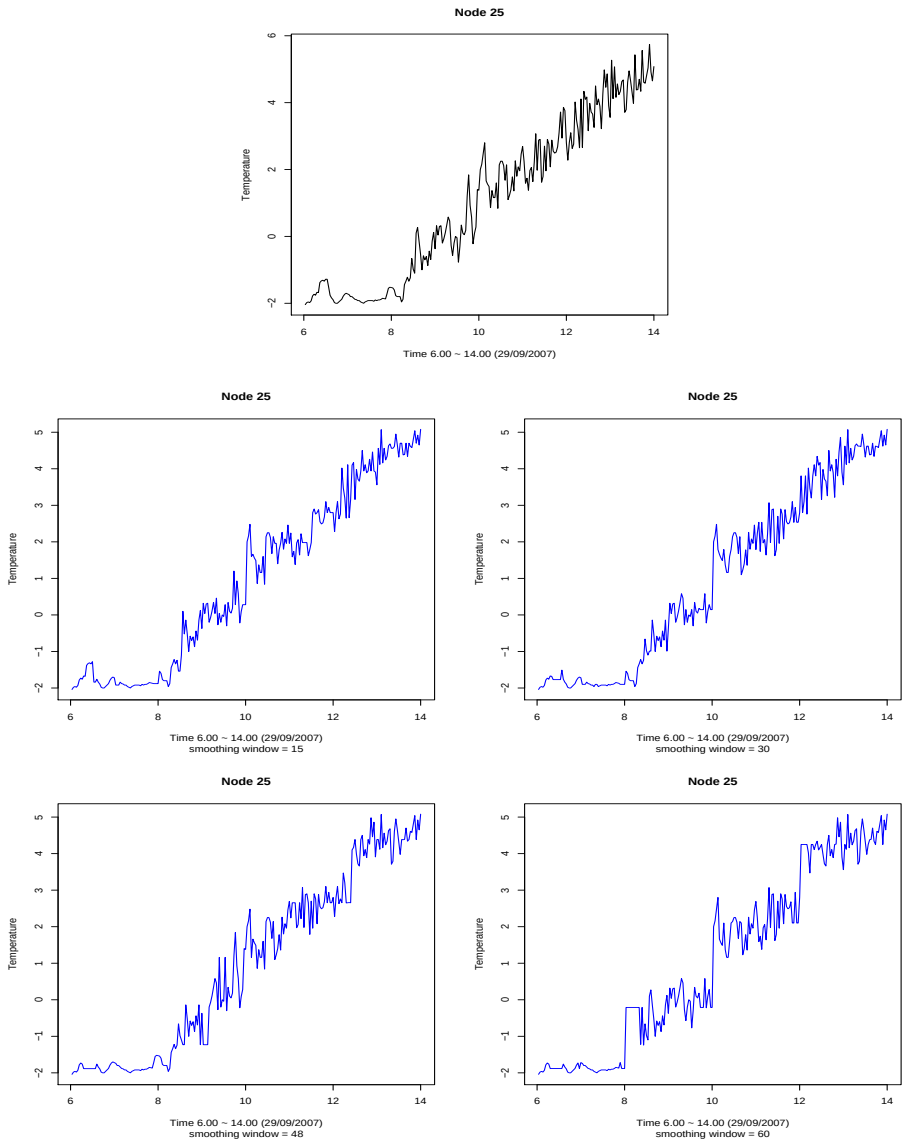


Figure 4.10: Original time series and corresponding smoothed time series using different sizes of smoothing window

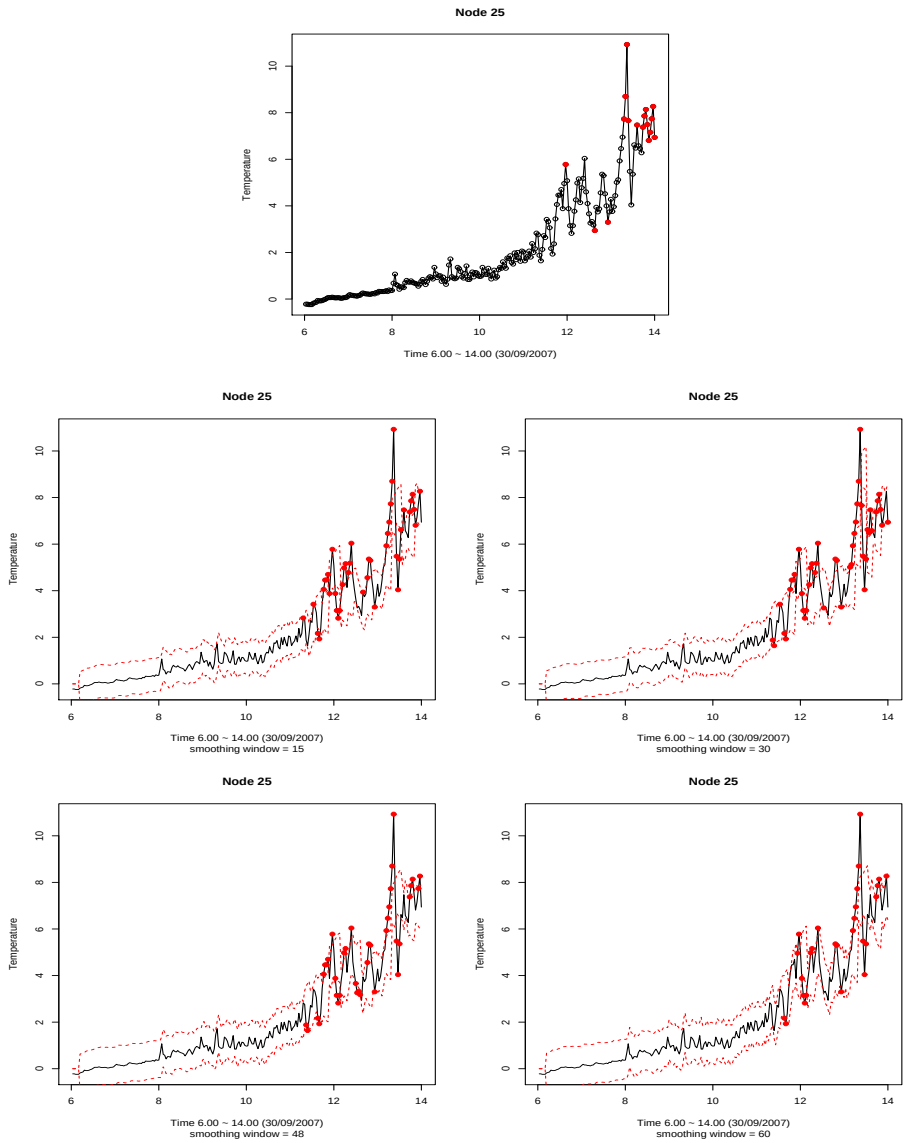


Figure 4.11: Labelled data using running average-based technique and temporal outliers detected by TOD for different sizes of smoothing window

Labelling Techniques	TOD	SW = 15	SW = 30	SW = 48	SW = 60
Running Average	DR	74.1573	73.03371	67.41573	57.30337
	FPR	10.62106	10.89109	9.450945	8.280828
Mahalanobis Distance	DR	82.85714	82.85714	82.85714	71.42857
	FPR	13.30472	13.47639	11.67382	10.12876
Density	DR	100	100	100	100
	FPR	14.83655	15.00419	13.24392	11.39983

Table 4.2: Detection rate (DR %) and false alarm rate (FPR %) of TOD on results of three labelling techniques for different sizes of smoothing window

Labelling Techniques	TOD	p=1	p=2	p=3	p=4
Running Average	DR	67.41573	73.03371	73.03371	74.1573
	FPR	9.810981	10.98110	10.98110	10.62106
Mahalanobis Distance	DR	82.85714	82.85714	82.85714	82.85714
	FPR	12.01717	13.56223	13.56223	13.30472
Density	DR	100	100	100	100
	FPR	13.57921	15.08801	15.08801	14.83655

Table 4.3: Detection rate (DR %) and false alarm rate (FPR %) of TOD on results of three labelling techniques for different orders of AR(p) model

be seen from Table 4.3, large p values results to higher detection rate, especially for running average-based labelling technique. The reason for this is that it uses many more previous observations to generate more reliable predicted values while reducing estimation errors. To ensure outlier detection results, we set the order of AR(p) model to 4, which is also consistent with the length of the sequence to distinguish between errors or events as mentioned before.

We further examine the effect of the confidence level for TOD. Figure 4.13 illustrates the results of applying TOD on density-based labelling technique for different values of confidence level. Table 4.4 shows the detection rate and false alarm rate of TOD on results of three labelling techniques for different values of confidence level. As it can be seen from Table 4.4, relatively small confidence level results in high detection rate and high false alarm rate on all three labelled

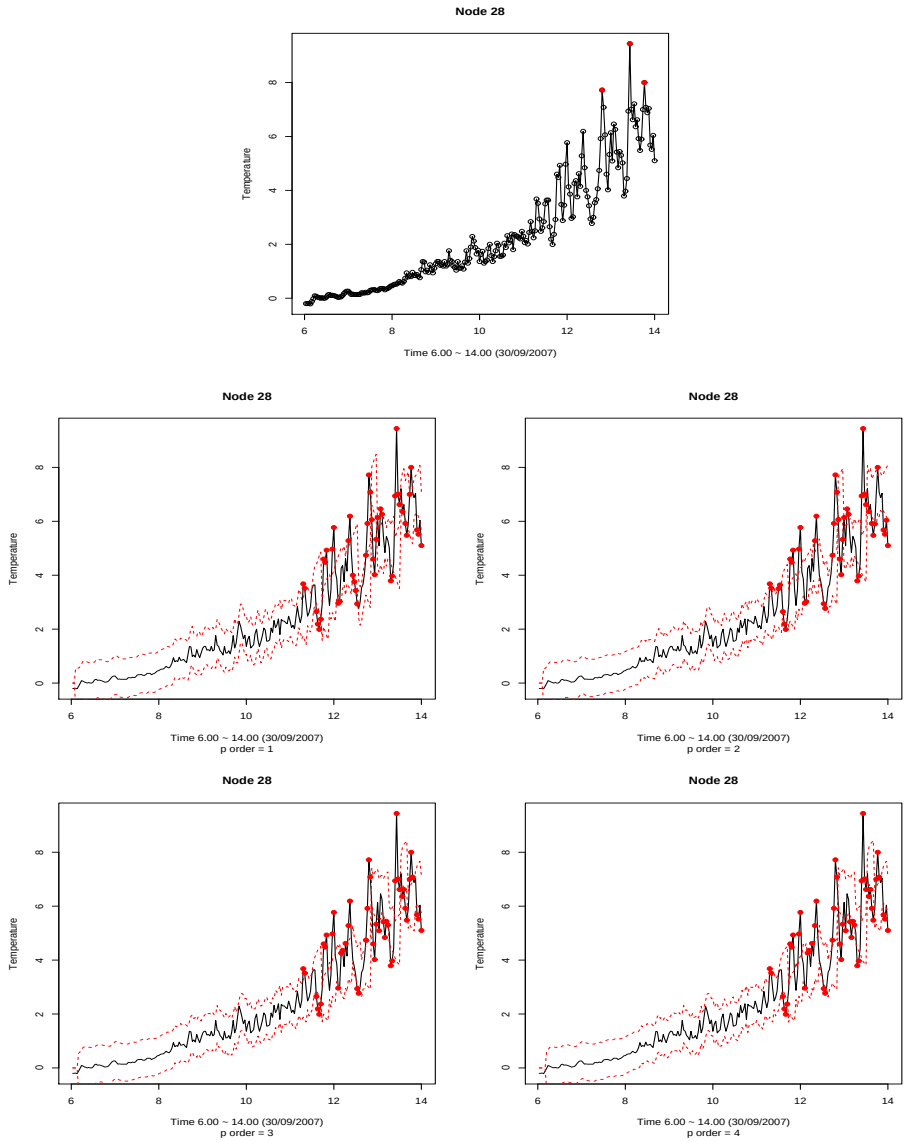


Figure 4.12: Labelled data using Mahalanobis distance-based technique and temporal outliers detected by TOD for different orders of AR(p) model

Labelling Techniques	TOD	CL=90%	CL=95%	CL=99%	CL=99.7%
Running Average	DR	78.65169	74.1573	51.68539	42.69663
	FPR	17.91179	10.62106	5.670567	2.520252
Mahalanobis Distance	DR	91.42857	82.85714	74.28571	62.85714
	FPR	20.34335	13.30472	7.124464	3.776824
Density	DR	100	100	100	100
	FPR	21.96144	14.83655	8.549874	4.945516

Table 4.4: Detection rate (DR %) and false alarm rate (FPR %) of TOD on results of three labelling techniques for different values of confidence level

datasets. On the contrary, large confidence level results in low false alarm rate and also low detection rate. This is due to the fact that using large confidence level has a large confidence level of predicted values and thus some outliers may be included in the confidence interval and are considered as normal. The small confidence level has high false alarm rate since it has a small confidence interval and considers some normal observations as outliers. Thus, there is a trade-off in choosing confidence level and performance of outlier detection. In the experiments, we set the value of confidence level to 95%, i.e., two standard errors.

We evaluate the performance of our TOD, compared with two kinds of prediction strategies. One is to at once predict values for all time instants, and the other is to sequently predict values for each time instant. Both of the two prediction strategies do not consider the actual observations at the new time series to be incorporated in the temporal correlation model. Figure 4.14 illustrates the results of applying TOD and two prediction strategies on results of running average-based labelling technique. Table 4.5 shows the detection rate and false alarm rate for temporal outliers detected by TOD and the two prediction strategies. As it can be seen from Figure 4.14 and Table 4.5, the two prediction strategies have bad performances although the prediction for each step results in high detection rate. This stems from in fact that they either use original time series or predicted values for prediction ignore how to deal with new observations identified as normal or outlier for future prediction and outlier detection. Our TOD achieves good detection rate while maintaining the false alarm low on three labelling techniques.

Table 4.6 shows the number of time sequences detected at nodes using TOD. We can seen that each node detects approximately same number of events at the nearby points. Due to the fact that TOD only uses local data of each node to detect changes in normal behavior of sensor data as well as the limitation

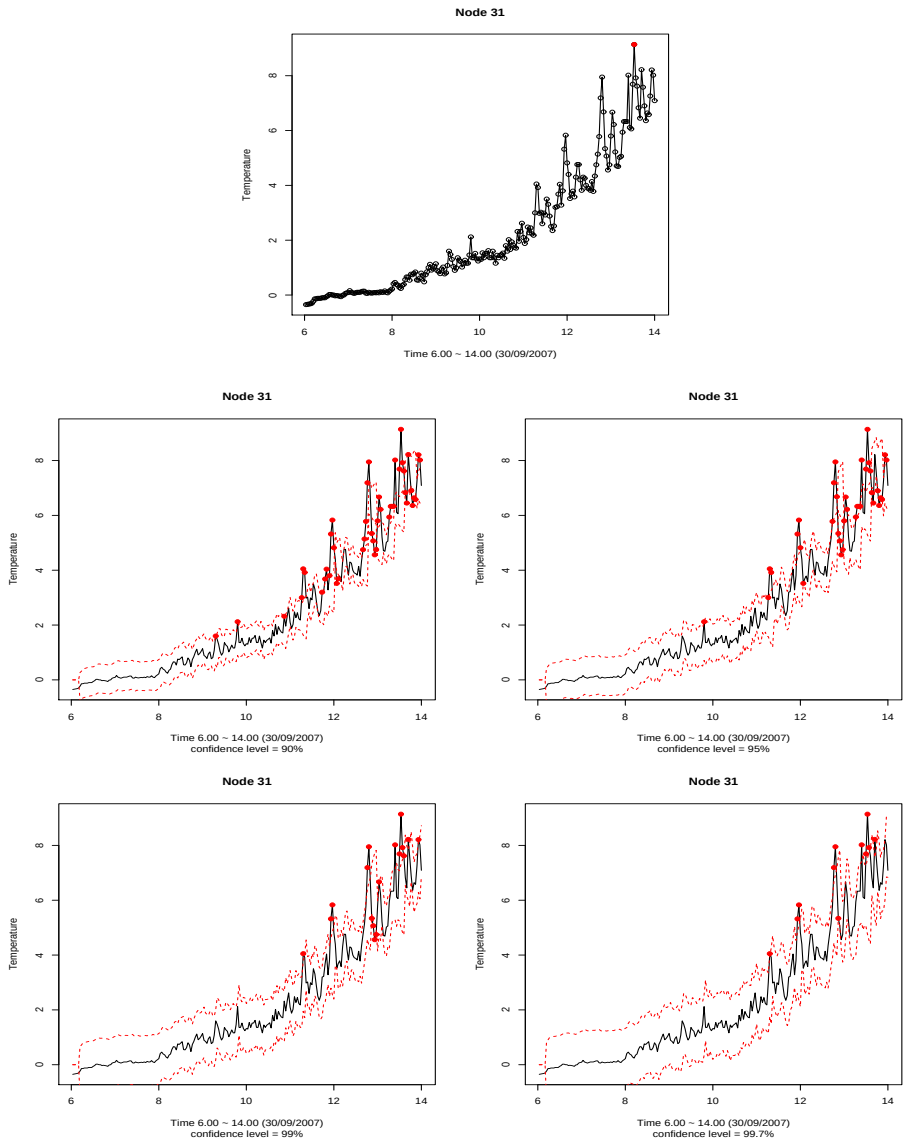


Figure 4.13: Labelled data using density-based technique and temporal outliers detected by TOD for different values of confidence level

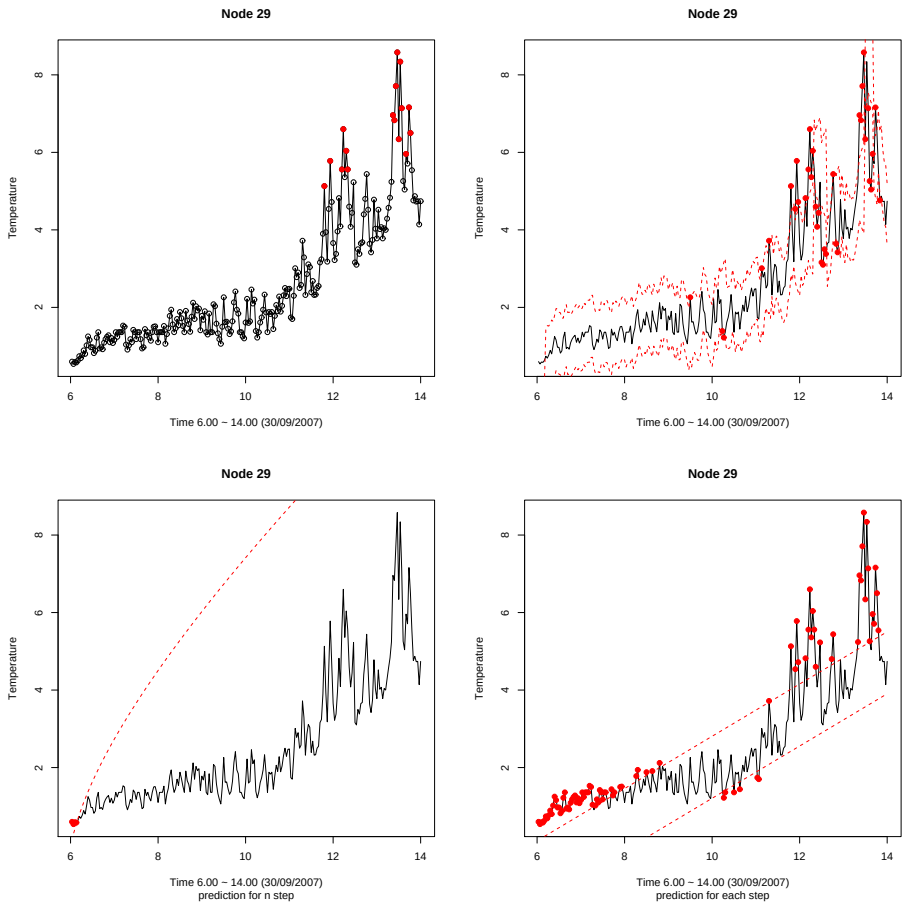


Figure 4.14: Labelled data using running average-based technique and temporal outliers detected by TOD and two kinds of prediction strategies

Labelling Techniques	Techniques	TOD	Prediction for all time instants	Prediction for each time instant
Running Average	DR	74.1573	1.123596	95.50562
	FPR	10.62106	5.220522	41.58416
Mahalanobis Distance	DR	82.85714	2.857143	100
	FPR	13.30472	4.978541	43.9485
Density	DR	100	14.28571	100
	FPR	14.83655	4.861693	45.26404

Table 4.5: Detection rate (DR %) and false alarm rate (FPR %) for temporal outliers using three labelling techniques for prediction strategies

Nodes	Number of outliers	Number of time sequences	Occurred points
Node 25	21	5	$\{173, 181, 216, 223, 232\}$
Node 28	23	5	$\{168, 202, 214, 222, 227\}$
Node 29	17	4	$\{186, 195, 221, 227\}$
Node 31	18	5	$\{202, 206, 218, 225, 233\}$
Node 32	12	2	$\{217, 222\}$

Table 4.6: Number of outliers and time sequences detected at different nodes of the small cluster using TOD

of length of sequence to distinguish between errors or events, the results of the occurrence of time sequence may not be reliable.

Spatial Correlation-Based Outliers

We here present results of spatial outliers detected by SROD and SPOD. First, to obtain a reliable variogram model in presence of lacking sufficient observation pairs, we take the method used in [114] and combine more observations at different time periods (each half hour) from 5 nodes. The fitted linear model (based on Equation 4.5) through the calculated variogram values is illustrated in Figure 4.15.

Using the fitted linear model, the weights of spatial neighbors for each node can be calculated for future value prediction and spatial outlier detection. Figure 4.16 and 4.17 illustrate the results of spatial outliers detected by SROD and SPOD, respectively. Table 4.7 shows number of spatial outliers detected by

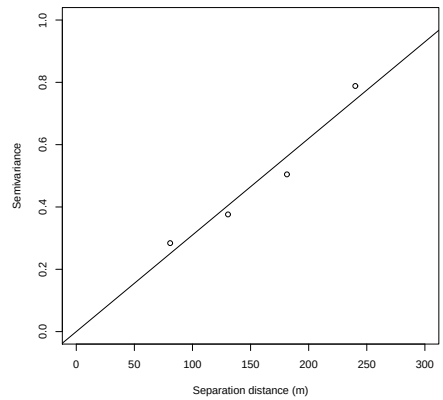


Figure 4.15: Calculated variogram of small cluster and the fitted linear model

Techniques	Node25	Node28	Node29	Node31	Node 32
SROD	14	21	24	20	8
SPOD	9	23	18	18	4

Table 4.7: Number of spatial outliers detected by SROD and SPOD at different nodes of the small cluster

SROD and SPOD at different nodes of the small cluster. Table 4.8 shows the detection rate and false alarm rate of SROD and SPOD on results of three labelling techniques. It can be seen that neither SROD nor SPOD achieves good detection accuracy while generating high false alarm. The reason for this is insufficient number of data points, i.e., only 5, which is not enough to indicate correct number of outliers in the labelling phase.

Finding spatial outliers requires a great deal of communication between nodes to exchange data and build the model. Such high communication and transmission overhead is not suitable for resource-constrained wireless sensor nodes. Therefore, performing spatial outlier detection alone is not our primary focus. As we will show in the later section, spatio-temporal outlier detection can not only reduce the communication overhead but also result in better detection rate and lower false alarm.

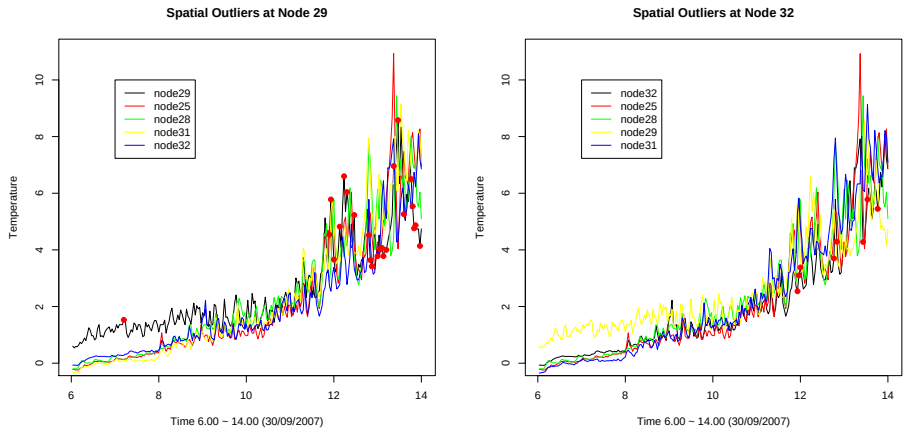


Figure 4.16: Spatial outliers detected by SROD

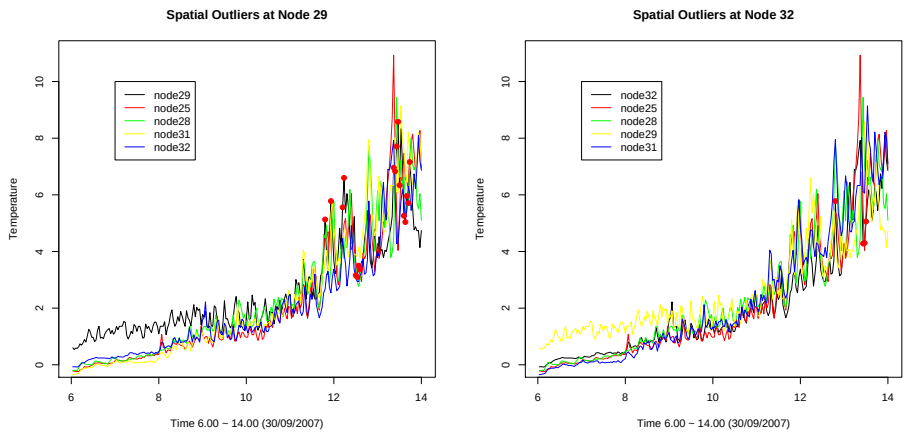


Figure 4.17: Spatial outliers detected by SPOD

Techniques	Labelling Techniques	Running Average	Mahalanobis Distance	Density
SROD	DR	5.172414	17.07317	2.8125
	FPR	7.180385	2.647413	8.636364
SPOD	DR	4.263372	15.52614	1.4732
	FPR	8.2457	4.251824	9.475458

Table 4.8: Detection rate (DR %) and false alarm rate (FPR %) for spatial outliers using three labelling techniques for SROD and SPOD

Techniques	Labelling Techniques	Running Average	Mahalanobis Distance	Density
TSOD	DR	26.59574	100	100
	FPR	1.898734	3.430962	3.511706
STGOD	DR	72.34043	100	100
	FPR	10.94033	15.39749	15.46823

Table 4.9: Detection rate (DR %) and false alarm rate (FPR %) for spatio-temporal temporal outliers using three labelling techniques for TSOD and STGOD

Spatio-Temporal Correlations-Based Outliers

We here evaluate the performance of TSOD and STGOD to detect spatio-temporal outliers. Figure 4.18 illustrates the results of applying our spatio-temporal outlier detection techniques TSOD and STGOD on dataset labelled by three labelling techniques. Table 4.9 shows the detection rate and false alarm rate of TSOD and STGOD on results of three labelling techniques. As it can be seen from Figure 4.18 and Table 4.9, STGOD achieves better performance compared with TSOD on running average-based labelled data dataset while TSOD has better performance than STGOD on Mahalanobis distance-based and density-based labelled dataset. Due to the fact that TSOD identifies an observation as an outlier only if this observation is a temporal outlier as well as a spatial outlier, thus it reduces the false alarm rate while causing lower detection rate, e.g., for running average-based labelling technique. The reason that TSOD has 100% detection rate on results of Mahalanobis distance-based and density-based labelling techniques is that there are very few observations labelled as outliers using the above

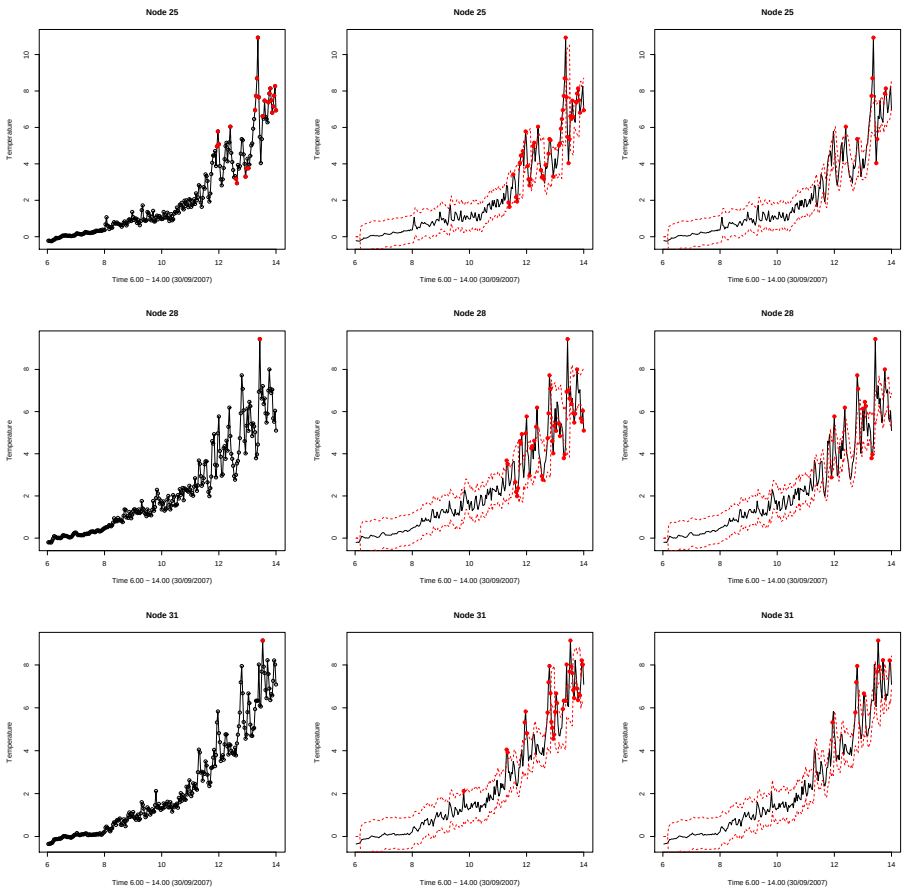


Figure 4.18: Labelled data using three labelling techniques (left column) and spatio-temporal outliers detected by STGOD (middle column) and TSOD (right column)

two labelling techniques.

4.6.3 Complexity Analysis

We further compare our statistical-based techniques in terms of communication overhead and computation and memory complexity. The communication complexity of our distributed techniques depends on the local transmission of modeling parameters and actual observations for modelling variogram. TOD requires no communication overhead due to local data analysis at each node. In SROD, each node sends it own observation at each time interval. The maximum communication overhead for each node in SROD, is therefore, $O(md)$, where m is the number of new observations to be classified and d is the dimension of observations. In SPOD, each node only transmits its parameters of temporal correlation modelling once at the initial detection phase and thus the maximum communication overhead in SPOD for each node is $O(nd)$, where n is the number of adjacent nodes. In TSOD, each node needs to send its own observation when an observation is identified as temporal outlier. Thus the maximum communication overhead of TSOD for each node is $O((m^{'} < m)d)$, where $m^{'}$ is the number of detected temporal outliers at each node. In STGOD, each node only transmits its parameters of temporal correlation modelling once at the initial detection phase and thus the maximum communication overhead in STGOD for each node is $O(nd)$.

The computational complexity in TOD concerns computation of the removal of trend, AR model and predicted values. The computational complexity of TOD mainly depends on fitting AR model, which can be represented as $O(p)$ by a linear optimization. Hence, the maximum computational complexity of each node in TOD is $O(mdp)$, where m is the number of original observations to be modelled and d is the dimension of observations. The computational complexity in SROD and SPOD mainly depends on fitting variogram and the computation of weights for spatial neighbors, which can be represented as $O(q)$ by a linear optimization. Hence, the maximum computational complexity of each node in SROD and SPOD is $O(mdq)$. The maximum computational complexity of each node in TSOD and STGOD is $O(md(p+q))$. When a gateway node is available to fit variogram and calculate weights for spatial neighbors, the computational complexity of each node will be significantly reduced.

The memory complexity of our techniques is mainly about keeping observations of the size of new observations in memory and it is represented as $O(md)$, where d is the dimension of observations and m is the number of new observations to be classified. Overhead of storing other parameters such as parameters of temporal correlation and spatial correlation modelling is negligible. Hence the

Techniques	Communication Complexity	Computational Complexity	Memory Complexity
TOD	–	$O(mdp)$	$O(md)$
SROD	$O(md)$	$O(mdq)$	$O(md)$
SPOD	$O(nd)$	$O(mdq)$	$O(md)$
TSOD	$O(((m^{'} < m)d))$	$O(md(p+q))$	$O(md)$
STGOD	$O(nd)$	$O(md(p+q))$	$O(md)$

Table 4.10: Complexity analysis of our outlier detection techniques for each sensor node

maximum memory complexity of each node for our techniques is $O(md)$. Table 4.10 summarizes these complexity.

4.7 Chapter Summary

In this chapter, we have proposed distributed and online statistical-based outlier detection techniques by taking advantage of spatial and temporal correlations to precisely detect outliers in WSNs. To cope with the problem of high complexity and overhead, we present solutions to efficiently model spatial and temporal correlations of sensor data to resource-constraint WSNs. By modelling spatio-temporal correlations, our proposed techniques can have a better understanding of internal data structure in space and time, precise identification of outliers, detection of changes in normal behavior of sensor data and further proceed to forecast observations by appropriately handling detected outliers. We compare performance of our techniques using real dataset as well as different labelling techniques. Extensive experimental results show that our outlier detection techniques achieve high detection accuracy and low false alarm, while keeping the computational complexity and communication overhead low.

Chapter 5

Spherical SVM-Based Outlier Detection Techniques for Wireless Sensor Networks

Assuming that an explicit probability distribution always exists for sensor data is not realistic and estimation of the corresponding distribution parameters is computationally expensive for WSNs. Data mining and machine learning-based approaches compute the similarity measure among data vectors without any assumption regarding specific data distribution. In this chapter, we introduce a quarter-sphere one-class SVM and efficiently utilize it to propose our distributed and online outlier detection techniques for multivariate sensor data in WSNs. We also take advantage of the theory of spatio-temporal correlations to identify outliers and sequently update the SVM-based model representing normal behavior of sensor data. Experimental results reveal that our proposed outlier detection techniques are able to precisely detect outliers and changes occurred in normal behavior of sensor data streams. They are, moreover, robust in terms of parameter selection.

5.1 Introduction

In Chapter 4, we have described how to efficiently capture spatial and temporal correlations of sensor data and utilize these modelled correlations to design our distributed and online statistical-based outlier detection techniques for WSNs. These mathematically justified spatial and temporal correlations-based techniques can effectively identify outliers and detect changes occurred in normal behavior of sensor data.

Modelling spatial and temporal correlations is a *parametric* statistical approach [71], which assumes that an explicit probability distribution exists in data and then estimates the required parameters of the probability distribution. However, sensor data in WSNs may not always comply with a specific statistical model due to the fact that internal spatial and temporal correlations existing among sensor data may not always be stable. Assuming existence of a statistical model for sensor data and estimation of its distribution parameters has two potential disadvantages: (i) the computational and memory complexity of the parameter estimation is rather expensive for WSNs, although there are some ways to lower this down, and (ii) estimated parameters of the assumed distribution may not be reliable as the distribution of sensor data may often change. Furthermore, capturing spatial and temporal correlations usually requires smoothed data without any abnormal, missing values, or misplaced data. In addition, the statistical approaches primarily deal with *univariate* data. Although they can also analyze multivariate data, computational and memory cost of doing so will be more expensive for WSNs.

To cope with these potential disadvantages of using statistical parametric approaches, some *non-parametric* outlier detection techniques [71] have been proposed by data mining and machine learning communities. Non-parametric outlier detection approaches have no assumption regrading specific statistical model for given data. Instead, they analyze data by computing the distance between data points as a *similarity measure* to discover the hidden and interesting knowledge in large datasets [118]. In this way, they avoid the assumption on possibly inappropriate data distribution as well as expensive computational and memory complexity of the parameter distribution estimation. Their unsupervised techniques can allow missing data or abnormal data existing during data analysis and have no strict requirements for the order of data arriving. In addition, these techniques usually are targeted to capture the similarity measure among *multivariate* data and even high dimensional data.

In this chapter, we use *quarter-sphere one-class SVM* [68] approach to identify outliers in multivariate sensor data and resource-constrained WSNs. Our proposed outlier detection techniques are designed to operate in a distributed

and online manner. In the process of detecting outliers, we also take advantage of the theory of spatio-temporal correlations to precisely detect global outliers and the change of normal behavior of sensor data. Furthermore, we present several strategies to update the SVM-based model that represents normal behavior of sensor data for further accurate outlier identification. Experiments with two synthetic datasets and real environmental dataset from the Grand St. Bernard [108] show that our proposed outlier detection techniques have the ability to precisely detect outliers and changes occurred in normal behavior of sensor data. Compared to a batch quarter-sphere SVM-based outlier detection technique [99], our proposed techniques represent robustness in terms of parameter selection.

The remainder of this chapter is organized as follows. Related work on using one-class SVM-based outlier detection in WSNs as well as in data mining and machine learning communities is described in Section 5.2. Principles of modelling the quarter-sphere one-class SVM classifier are addressed in Section 5.3. How to lower down the complexity of traditional quarter-sphere one-class SVM classifier to fit requirements of WSNs is addressed in Section 5.4. Our proposed quarter-sphere SVM-based outlier detection techniques are presented in Section 5.5. Experimental results and performance evaluation of our techniques are reported in Section 5.6. Finally this chapter is concluded in Section 5.7.

5.2 Related Work

SVM-based techniques come from the family of classification-based techniques in data mining and machine learning communities. Classification-based techniques learn a classifier using data vectors in the training phase and classify an unseen instance into one of the learned classes in the testing phase. SVM-based techniques specifically separate the data belong to different classes by fitting a hyperplane, which produces a maximal margin in a high-dimensional data space. They are commonly used for the purpose of outlier detection due to the fact that they have three main attracting advantages including: (i) they do not require an explicit statistical model and parameter estimation, (ii) they use an optimum solution to produce a more reliable normal boundary to precisely distinguish between normal data and outliers, and (iii) they avoid the curse of data dimensionality problem [118] for computing the similarity measure among data vectors.

However, traditional SVM-based outlier detection techniques suffer from their two disadvantages: (i) they require error-free or labelled data for training, and (ii) they require a computationally expensive quadratic optimization. One-class (*unsupervised*) SVM-based techniques can solve the first disadvantage, as they can model normal behavior of the unlabelled data while ignoring the anomalies

existing in the training set. Their main idea is to use a non-linear function to map the data vectors collected from the original *input space* to a higher dimensional space called *feature space*. Then a decision boundary of normal data is found, which encompasses the majority of the data vectors in the feature space. Those data vectors falling outside the normal boundary are classified as outliers. To this end, Scholkopf et al. [106] have proposed a *hyperplane-based* one-class SVM, which identifies outliers by fitting a hyperplane from the origin. The data vectors near the origin are declared as outliers. Tax et al. [122] have proposed a *hypersphere-based* one-class SVM, which identifies outliers by fitting a hypersphere with a minimum radius. The data vectors falling outside of the hypersphere are declared as outliers. These techniques still require a computationally expensive quadratic optimization. In order to reduce high computational cost of the quadratic optimization, Campbell et al. [23] have formulated a *linear* programming approach for the hyperplane-based SVM in [106], which is based on attracting the hyperplane towards the average of the distribution of mapped data vectors. Laskov et al. [68] have extended work in [122] by proposing a *quarter-sphere* one-class SVM, which converts the quadratic optimization problem to a *linear optimization* problem by fitting a hypersphere centered at the origin, and thus reducing the effort of computational complexity of learning the normal boundary.

Rajasegarar et al. [99] use the quarter-sphere one-class SVM proposed in [68] to present a distributed outlier detection technique for WSNs. They achieve approximately similar outlier detection results compared with a centralized approach. In their technique, each node analyzes sensor data in an offline manner only after all observations are collected within a day, which obviously causes a considerable outlier detection delay. This is not suitable for detecting outliers in critical real-time applications of WSNs. Moreover, each node communicates just only its own radius of the quartersphere with its neighboring nodes to identify global outliers while ignoring other important parameters such as mean or standard deviation. This impacts the reliability of final outlier detection results. Also, this technique models the quarter-sphere SVM in the feature space using centered kernel functions, which has high computational and memory complexity.

Although we use the same quarter-sphere one-class SVM as presented in [68], our proposed outlier detection techniques simplify the process of modeling quarter-sphere SVM for WSNs and identify outliers in an online manner. They are also able to detect changes occurred in normal behavior of sensor data in real-time, which significantly reduces the detection delay. We consider not only the modelled radius but also other related parameters such as mean and standard deviation, which contribute to provide more reliable outlier detection results. Additionally, we take advantage of the theory of temporal correlation to detect the change of normal behavior occurred between two continuous time windows.

5.3 Principles of Modelling Quarter-Sphere One-Class SVM

In this section, we describe principles of modelling the quarter-sphere one-class SVM proposed in [68] for multivariate data vectors. This SVM-based technique fixes the center of mapped data vectors in the feature space at the origin and thus converts the quadratic optimization problem to the linear optimization problem during modeling the quarter-sphere SVM classifier. The geometry of quarter-sphere one-class SVM-based approach is shown in Figure 5.1. The general process of modeling the quarter-sphere SVM classifier for multivariate data vectors is addressed below.

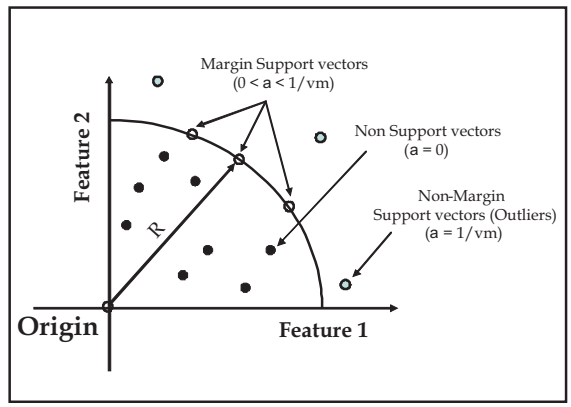


Figure 5.1: Geometry of the quarter-sphere formulation of one-class SVM [68]

Considering that m data vectors $\{x_i \epsilon \Re^d, i = 1, \ldots, m\}$ of d variables in the input space are mapped into the feature space using some non-linear mapping function ϕ, the quarter-sphere SVM aims at enclosing majority of mapped data vectors $\phi(x_i)$ in the feature space by fitting a quartersphere centered at the origin with a minimum radius R. Thus, the optimization problem in this quarter-sphere SVM classifier is formalized as follows:

$$\min_{R \epsilon \Re, \xi \epsilon \Re^m} \quad R^2 + \tfrac{1}{vm} \sum_{i=1}^{m} \xi_i \tag{5.1}$$

$$subject\ to: \|\phi(x_i)\|^2 \leq R^2 + \xi_i, \xi_i \geq 0, i = 1, 2, \ldots m$$

where $\upsilon \in (0,1)$ is a parameter that specifies the tradeoff between the quarter-sphere volume and outliers. This implies that υ represents the fraction of mapped data vectors that can be outliers. Slack variables denoted as $\{\xi_i : i = 1, 2, \ldots m\}$ are variables indicating soft boundary of the quarter-sphere SVM allowing some of mapped data vectors to fall outside the quarter-sphere. Accordingly, the Lagrange function for this optimization can be presented as:

$$L = \ R^2 - \sum_{i=1}^{m} \alpha_i (R^2 - \|\phi(x_i)\|^2 + \xi_i) - \sum_{i=1}^{m} \beta_i \xi_i + \frac{1}{\upsilon m} \sum_{i=1}^{m} \xi_i \tag{5.2}$$

where $\alpha_i \geq 0$, $\beta_i \geq 0$ for all $i = 1, 2, \ldots, m$ are the Lagrangian multipliers. Taking the zero derivatives of L with respect to R and ξ_i result in:

$$\frac{\partial L}{\partial R} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i = 1, \tag{5.3}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad \alpha_i = \frac{1}{\upsilon m} - \beta_i \tag{5.4}$$

By having $\alpha_i \geq 0$, $\beta_i \geq 0$ in Equation 5.4, we obtain $0 \leq \alpha_i \leq \frac{1}{\upsilon m}$. Substituting Equation 6.2 and Equation 5.4 into Equation 5.2 results in:

$$L = \sum_{i=1}^{m} \alpha_i (\|\phi(x_i)\|^2) = \sum_{i=1}^{m} \alpha_i (\phi(x_i) \cdot \phi(x_i)) \tag{5.5}$$

where $\|\phi(x_i)\|^2 = (\phi(x_i) \cdot \phi(x_i))$ is the *inner product* of the mapped data vector $\phi(x_i)$. This inner product indicates the similarity measure between $\phi(x_i)$ and $\phi(x_i)$ in the feature space. It then can be replaced by a kernel function $k(x_i, x_i)$, which is a commonly used tool to compute the inner product of any of two vectors in the feature space by the original data attributes [119]. Hence, the dual formulation of Equation 5.1 will become:

$$\min_{\alpha \epsilon \Re^m} \quad - \sum_{i=1}^{m} \alpha_i k(x_i, x_i) \tag{5.6}$$

$$subject\ to: \sum_{i=1}^{m} \alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{\upsilon m}, i = 1, 2, \ldots m$$

Now this dual problem in Equation 5.6 becomes a linear optimization problem, and $\{\alpha_i\}$ values can be effectively obtained by linear optimization techniques such

as simplex method or the interior point method [80]. Consequently, the data vectors can be classified depending on the results of $\{\alpha_i\}$. As shown in Figure 5.1, the data vectors with $\alpha_i = 0$ falling inside the quarter-sphere are called *non support vectors*, which are considered as normal data. The data vectors with $\alpha_i > 0$ are called *support vectors*. They determine the computational complexity and accuracy of the quarter-sphere SVM. Support vectors with $0 < \alpha_i < \frac{1}{\upsilon m}$ falling on the quarter-sphere are called *margin support vectors*. Their distances to the quarter-sphere center indicate the minimum radius R, which can be obtained by $R^2 = k(x_i, x_i)$ for any margin support vectors x_i. These margin support vectors are also considered as normal. Support vectors with $\alpha_i = \frac{1}{\upsilon m}$ falling outside the quarter-sphere are called *non-margin support vectors*. They are considered as outliers since their distances to the quarter-sphere center are larger than R.

The dual problem shown in Equation 5.6 indicates its solution only depends on the inner products of mapped data vectors. This poses a problem when a distance-based kernel function, e.g., radial basis function (RBF) [132], is used to map the original data vectors, where the inner products of mapped data vectors are equal and thus no meaningful solution to the dual problem can be found. In order to alleviate this problem, the mapped data vectors needs to be centered in the feature space. This can be done by subtracting the mean $\mu = \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)$ from all mapped data vectors, i.e., $\hat{\phi}(x_i) = \phi(x_i) - \mu$ [106]. The *centered kernel matrix* K_c then can be easily computed in terms of the kernel matrix $K = k(x_i, x_j) = (\phi(x_i) \cdot \phi(x_j))$ using $K_c = K - 1_m K - K 1_m + 1_m K 1_m$, where 1_m is an $m \times m$ matrix with all values equal to $\frac{1}{m}$. After centering in feature space, all inner products of mapped data vectors are not longer equal and the dual problem of the quarter-sphere formulation can be solved.

5.4 Fitting Quarter-Sphere One-Class SVM Modelling to Resource-Constraint WSNs

Modelling the quarter-sphere one-class SVM involves the generation of $m \times m$ kernel matrix K in the feature space and the transformation of central kernel matrix K_c, which brings plenty of computational and memory complexity for WSNs. In order to reduce the resource cost of SVM modelling, we model the quarter-sphere SVM in the *input space* and fix the center of quartersphere at the origin. For doing so, raw sensor data first may need to be transformed to achieve a more symmetric data distribution. This work can be done using the *Box-Cox method* [129], which is commonly used for data transformation. This

transformation is essentially a *log-transformation* defined for positive data vectors. In case of negative values, a constant can be added to make them positive. Afterwards, the data vectors can be centered at the origin using the *autoscaling* (*z-transformation*) [129], which is the most used data preprocessing approach. Considering the fact that sensor observations collected by different type of sensors may have different scales, autoscaled values can also make all variables fall in the same scale and avoid having a variable with larger magnitudes masking the influence of variables with smaller magnitudes. For a data vector x_i, its *autoscaled value* is formulated as $x_i^{'} = (x_i - \mu)/\sigma$, where μ is the mean of the attribute values and σ is corresponding standard deviation. Since autoscaled values may be sensitive to outliers, we replace the arithmetic mean by the *median* and replace the standard deviation by the *median absolute deviation* (MAD), which are both more robust against extreme high and low values [129].

After this data preprocessing phase, the data vectors are centered at the origin in the input space. This is to lower down the computational and memory complexity of modelling quarter-sphere SVM. Consequently, the dual formulation of Equation 5.6 in the input space will be simplified as:

$$\min_{\alpha \epsilon \Re^m} \quad -\sum_{i=1}^{m} \alpha_i (x_i^{'} x_i^{'T}) \tag{5.7}$$

$$subject\ to: \sum_{i=1}^{m} \alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{vm}, i = 1, 2, \ldots m$$

where $x_i^{'} x_i^{'T}$ represents the distances of autoscaled data vectors in the input space from the origin. These distances indicate the similarity measure between data vectors and they will be used to model the quarter-sphere SVM in the input space.

The quarter-sphere one-class SVM-based technique proposed in [68] essentially separates data vectors and outliers in an offline manner. After modelling the quarter-sphere SVM, the outliers are identified depending on their corresponding $\{\alpha_i\}$ values. This technique does not provide online detection of outliers for new arriving data vectors. To achieve online outlier detection for WSNs, we present a basic decision function to determine whether a new arriving sensor observation x is outlier by modelling the quarter-sphere SVM in the input space. According to Equation 5.7, the decision function can be computed as:

$$f(x) = sgn(R^2 - d(x^{'})^2) = sgn(R^2 - x^{'} x^{'T}) \tag{5.8}$$

where the square radius R^2 of the quartersphere plays the role of a threshold. It

can be computed by the inner product of any margin support vectors in the input space. The observations with a negative value are classified as outliers since their square distances from the origin in the input space are larger than R^2.

The inner products of original data vectors in the input space are easily computed by $x_i^{'} x_i^{'T}$. For mapped data vectors in the feature space, however, their distances from the origin are more difficult to be specified, especially for the distance-based kernel function due to its centered kernel matrix. To clearly express this, we use the above data preprocessing for raw sensor data and then map them into the feature space. For a new observation x, the decision function indicating whether it is determined as normal or outlier by the modelled quarter-sphere SVM in the feature space can be represented as follows:

$$f(x) = sgn(R^2 - d(\phi(x^{'}))^2) = sgn(R^2 - \|\phi(x^{'}) - \frac{1}{m}\sum_{i=1}^{m}\phi(x_i^{'})\|^2)$$

$$= sgn(R^2 - (k(x^{'}, x^{'}) + \frac{1}{m^2}\sum_{i=1}^{m}k(x_i^{'}, x_j^{'}) - \frac{2}{m}\sum_{i=1}^{m}k(x^{'}, x_i^{'})) \qquad (5.9)$$

As it was shown in Equation 5.9, the decision function used in the feature space is computationally more expensive compared to the decision function in the input space (based on Equation 5.8).

5.5 Spherical SVM-Based Outlier Detection Techniques

We have described in the previous section how to efficiently model the quarter-sphere SVM in the input space and provide the decision functions to determine whether new observations are normal or outliers in the both data spaces. Here we use the modelled quarter-sphere SVM to identify outliers and detect the change of normal behavior of sensor data in an online manner.

We consider the same network topology illustrated in Figure 4.6 in Chapter 4. At a time instant t, $x(s_1, t), \ldots, x(s_n, t)$ denote the data vector measured at nodes $s_1, \ldots s_n$, respectively. Each data vector at the corresponding node is composed of multiple attributes $x^l(s_i, t)$, where $x^l(s_i, t) = \{x^l(s_i, t) : i = 1 \ldots n, l = 1 \ldots d\}$ and $x(s_i, t)\ \epsilon\ \Re^d$.

Our spherical SVM-based online outlier detection technique (SOOD) enables each node to determine whether its every new observation is normal or outlier in

real-time using Equation 5.8 and 5.9. Specifically, each node s_i models its quarter-sphere SVM for sensor data during a time interval, in which m observations are made. As a result, each node obtains its radius R_i of the modelled quartersphere together with the median and the MAD. The R_i as a threshold represents the normal boundary of sensor observations in the time window. The median and the MAD are used to center the data vectors at the origin. They all are important parameters of the modelled quarter-sphere SVM. As we stated in Chapter 4, sensor observations are more likely to be temporally correlated at the certain time period in two consecutive days, each node thus can use the quarter-sphere SVM modelled at a short time window in the previous day to determine its new observations as normal or outlier at the corresponding time window in the next day. It implies that the size of m is based on actual monitoring process. On the other hand, each node often is required to identify outliers in a more global perspective. To achieve that, each node needs to communicate the R_i as well as the parameters of the median and the MAD with its spatial neighbors to cooperatively identify outliers. This is because sensor observations measured are more likely to be spatially correlated. The main steps of SOOD are:

- *Step 1.* Each sensor node s_i models its quarter-sphere SVM for m observations during a time interval and then obtains the threshold R_i as well as the corresponding median and the MAD parameters of the centered data. Then the local outliers can be determined in real-time by comparing R_i and the distances between new observations and the origin using Equation 5.8 and 5.9.

- *Step 2.* Each node s_i communicates its parameters R_i as well as the median and the MAD with its spatial neighbors by the radio transmission. This implies that each node communicates $1 + 2d$ elements over the network, where d is the dimension of a sensor observation.

- *Step 3.* Each node s_i combines R_i as well as the median and the MAD parameters from its neighbors together with its own R_i, median and MAD. The merged parameters are denoted as R_g, representing the global median and the global MAD, which are uniform for all nodes in the local region.

- *Step 4.* Each node s_i uses its merged parameters to online determine global outliers for their new observations. For a new observation x, the decision functions indicating whether it is a global outlier in the input space as well as in the feature space can be defined as:

$$f(x) = sgn(R_g^2 - x^{'} x^{'T}) \tag{5.10}$$

$$f(x) = sgn(R_g^2 - (k(x^{'}, x^{'}) + \frac{1}{m^2}\sum_{i=1}^{m} k(x_i^{'}, x_j^{'}) - \frac{2}{m}\sum_{i=1}^{m} k(x^{'}, x_i^{'})) \quad (5.11)$$

SOOD technique scales well with increase of number of nodes due to its distributed processing nature. It can update the merged parameters by communicating among spatially neighboring nodes at the end of each time interval. It has low communication overhead and computational complexity and does not need to transmit any actual observations between sensor nodes only the parameters. SOOD can also be extended to identify various types of outliers in real-time using the same strategies described in Chapter 4 and detect the change of normal behavior occurred in two consecutive time windows.

SOOD does not update the existing quarter-sphere SVM model until the end of the entire time interval. It indeed can detect the change of normal behavior between two consecutive time windows when most of observations measured in the next time window are detected as outliers. However, it can not detect changes in normal behavior of data within the time window or adapt to new behavior of sensor data. Therefore, SOOD may suffer from a possibly high rate of false alarm since new (yet normal) observations are detected as outliers. In order to alleviate this problem, our three outlier detection techniques incorporate new arrived observations, update the modelled quarter-sphere SVM for more reliable outlier detection, and detect the change of normal behavior of sensor data. Our update strategies include updating the model (i) at each time interval, (ii) after a fixed-size time window, and (iii) depending on the previous decision results.

5.5.1 Spherical SVM-Based Instant Outlier Detection Technique (SIOD)

Here we present an instant outlier detection technique (SIOD), which instantly updates the quarter-sphere SVM using a *sliding window* [42]. This update allows the method to deal with possible smooth variations in the structure of data streams [26]. The simplest method of updating our SVM-based model over time is to compute the radius of one-class quartersphere at each time instant as well as the median and the MAD. This implies that these parameters should be recomputed after every new observation is inserted into the sliding window and the outlier detection process is completed. During each update, the current observation is inserted to the sliding window while the oldest observation is removed from it. The median and the MAD of the new sliding window are then re-calculated. The new radius $R^{'}$ can be subsequently derived by solving the linear optimization

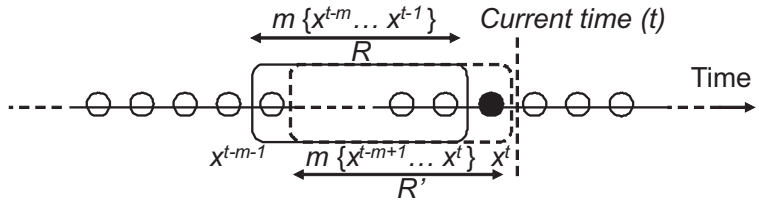


Figure 5.2: Update policy of SIOD's model. Circles represent sensor observations. The sliding window is composed of the last m observations. The black dot represents the observation identified at current time t.

problem. This procedure is repeated for evolving sliding window of a fixed size. Once the radius $R^{'}$ as well as the median and the MAD of a node are updated at each time instant, the node locally broadcasts these new parameters to its spatially neighboring nodes. When receiving the radii from all of its neighbors, each node combines these parameters with its own to compute the new global radius $R_g^{'}$ as well as the new global median and MAD, which are further used to identify its next observation as normal or outlier. Figure 5.2 illustrates the update policy of SIOD's model. The corresponding pseudocode for SIOD is shown in Table 5.1, which includes three processes, i.e., modelling SVM, outlier detection, and updating SVM model. One can note that the size of sliding window can be tuned using a priori knowledge about the data and cam be adjusted in accordance with the dynamics of the monitored process [26]. We use the same m as the size of sliding window as SOOD.

5.5.2 Spherical SVM-Based Fixed-Size Time Window-Based Outlier Detection Technique (SFTWOD)

It can be clearly seen that SIOD is computationally expensive and has high communication overhead due to updating the normal model every time a new observation is arrived. Thus, a slight modification to SIOD is to identify each observation upon being measured but update the model at a fixed-size time interval. This means that the sliding window will be freezed during measuring the next n ($n \ll m$) observations. Identification of outliers for these n observations is performed using the previously defined normal model. This modification effectively reduces computational and communication complexity as well as the time of recomputing the radius of one-class quartersphere. Since each of the n observations will be classified as normal or outlier upon arrival, it ensures that there is no delay in outlier detection itself.

```
1 procedure ModellingSVMProcess()
2   each node models the quarter-sphere SVM;
3   each node locally broadcasts the modeled quarter-sphere R_i as well as the median
    and the MAD to its spatially neighboring nodes;
4   each node then computes the global R_g as well as the global median and MAD;
5   initiate OutlierDetectionProcess(R_g, the global median and MAD);
6   return;
7 procedure OutlierDetectionProcess(R_g, the global median and MAD)
8   when x(t) arrives
9      compute d(x);
10      if (d(x) > R_g)
11          x(t) indicates an outlier;
12     else
13          x(t) indicates a normal observation;
14     endif;
15     initiate UpdatingSVMProcess(x(t));
16     set t ← t + 1;
17  return;
18 procedure UpdatingSVMProcess(x(t))
19  update the sliding window: the oldest observation x(t − m) is removed
    and replaced by x(t);
20  update the R_i as well as the median and the MAD for the sliding window;
21  locally broadcast the updated R_i and the median and the MAD
    to its neighboring nodes;
22  update the global R_g as well as the global median and MAD;
23  return;
```

Table 5.1: Pseudocode of SIOD

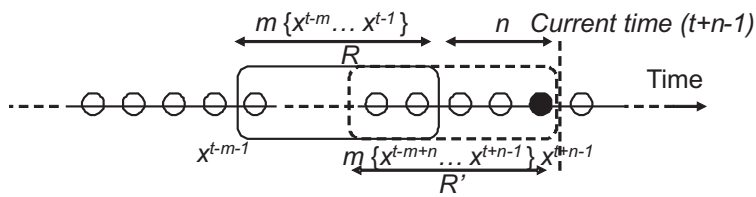


Figure 5.3: Update policy of SFTWOD's model. The sliding window is updated at each n observations.

1-6 procedure **ModellingSVMProcess**()

7-14 procedure **OutlierDetectionProcess**(R_g, the global median and MAD)
15 if (t % n == 0)
16 initiate **UpdatingSVMProcess**($x(t) \dots x(t+n-1)$);
17 set $t \leftarrow t+n$;
18 endif;
19 return;

20 procedure **UpdatingSVMProcess**($x(t) \dots x(t+n-1)$)
21 update the sliding window: the older observations $x(t-m) \dots x(t-m+n-1)$ is removed and replaced by $x(t) \dots x(t+n-1)$;
22 update the R_i as well as the median and the MAD for the sliding window;
23 locally broadcast the updated R_i and the median and the MAD to its neighboring nodes;
24 update the global R_g as well as the global median and MAD;
25 return;

Table 5.2: Pseudocode of SFTWOD

During each update, the previous n observations are inserted to the sliding window while the oldest n observation are removed from it. Figure 5.3 illustrates the update policy of SFTWOD. The corresponding pseudocode for SFTWOD is shown in Table 5.2. In general, n should be much smaller than m (m being the size of the sliding window) because large n will result in missing small behavioral changes in the dataset. Since environmental changes occur gradually, choosing a big n will lead to failure of the outlier detection techniques. On the contrary, if *n=1*, SFTWOD becomes like SIOD, which has high computational cost because it frequently updates the normal boundary at each time interval. Note that this technique differs from the batch technique proposed in [99], as the latter does not consider any possible relationship between data of two consequent sliding window and independently updates the normal model at every time window. Ignoring this relationship in [99] causes the detection delay due to waiting for the entire sliding window.

5.5.3 Spherical SVM-Based Adaptive Outlier Detection Technique (SAOD)

The update policy of the above-mentioned techniques introduce relatively high complexity and communication overhead due to the fact that each node is required

to frequently update its model. On the other hand, although they instantly detect outliers and the change of normal behavior occurred in sensor data, it is more likely that accuracy of detecting outliers decreases by having more outliers inside the sliding window. For the sake of energy efficiency and computational simplicity as well as the reliability of outlier detection results and detection of the changing distributional behavior of sensor data, we introduce an adaptive outlier detection technique (SAOD), which is based on use of relationship between the previous decision results and the modelled quarter-sphere SVM.

The performance of outlier detection using unsupervised SVM-based models mainly depends on the choice of the parameter ν, which controls the fraction of data vectors that can be outliers. This parameter actually denotes the *upper bound* on the fraction of detected outliers in a dataset [106], which indicates the maximum number of outliers allowed by the SVM model. The value of ν is usually chosen based on the a priori knowledge of the data and its normal behavior [26]. Thus we can mildly come to the conclusion that if the fraction of outliers detected by the quarter-sphere SVM model exceeds the given upper bound (ν) within a time period, it indicates that a new normal behavior is emerged and the previously modelled SVM is not suitable to represent the current normal behavior of sensor data any more and it needs to be updated. Therefore, we can set different time intervals to check if any new normal behavior exists. To reduce the computational and communication cost of updating the SVM model and being able to compare the performance of our outlier detection technique with the batch technique of [99], we will check if the fraction of outliers detected by the quarter-sphere SVM model exceeds the given upper bound (ν) at the end of sliding window (m).

Unlike the two previous techniques, in SAOD each node does not update the model (R) after a new observation is detected as normal or outlier. It only updates its own median and MAD values and incorporates the new observation into the sliding window while removing the oldest observation from it. These parameters are used to identify the next arriving observation as normal or outlier without a need for further communication among nodes. One should note that all new observations, regardless of being detected as normal or outlier, can be incorporated into the sliding window due to the fact that the parameter ν of unsupervised SVM allows anomalous observations in the training set. Moreover removing anomalous observations would bias the normal boundary [26]. Each node will check if the number of detected outliers exceeds the given upper bounder (ν). If so, a new normal behavior is detected and SVM model should be updated. After updating the quarter-sphere SVM, each observation can be labeled as normal or outlier according to Equation 5.10 and 5.11.

SAOD enables to detect outliers robustly using sequential observations and

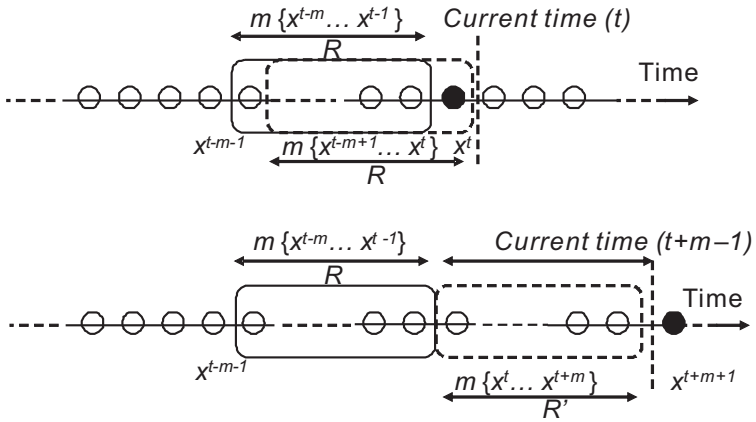


Figure 5.4: Update policy of SAOD's model. SAOD identifies outliers in real-time. Finally SAOD would update the SVM-based model and re-identify outliers if the number of detected outliers exceeds the given upper bounder ν.

1-6 procedure **ModellingSVMProcess**()

7-14 procedure **OutlierDetectionProcess**(R_g, the global median and MAD)
15 initiate **UpdatingSVMProcess**($x(t)$);
16 set $t \leftarrow t + 1$;
17 if (m data observations are collected)
18 if (the number of detected outliers > the given upper bound (ν))
19 update the SVM model for $(x(t - m + 1) \dots x(t))$ for outlier detection;
20 endif;
21 endif;
22 return;

23 procedure **UpdatingSVMProcess**($x(t)$)
24 update the sliding window: the oldest observations $x(t - m)$ is removed and replaced by $x(t)$;
25 update the median and the MAD for the sliding window;
26 return;

Table 5.3: Pseudocode of SAOD

also detect the change of normal behavior of sensor data. It helps to recognize the previously detected outliers as the indication of a new normal behavior and reduce the false alarm rate occurred in SOOD. Moreover, SAOD is communication efficient while having significantly less computational time. Figure 5.4 illustrates the update policy of SAOD. The corresponding pseudocode modification for SAOD is shown in Table 5.3.

5.6 Experiments

This section describes performance evaluation of our SOOD, SIOD, SFTWOD and SAOD techniques compared to SBOD presented earlier in Rajasegarar et al. [99]. The goals of our evaluation are (i) to test the accuracy of our distributed and online outlier detection techniques and their robustness in terms of parameters selection and (ii) to investigate impact of different labelling techniques described in Chapter 3 on performance of outlier detection techniques.

5.6.1 Experimental Datasets

In our experiments, we use two synthetic datasets as well as a real dataset gathered from the Grand St. Bernard [108]. The synthetic data used here is similar to the one used in [104]. We consider a sensor sub-network consisting of 7 sensor nodes, which are within radio transmission range of each other.

The first 2-D synthetic dataset is composed of 200 data vectors for each node having a mixture of three Gaussian distributions with uniform outliers. The mean value of this dataset is randomly selected from (0.32, 0.35, 0.38) while the standard deviation is set to be 0.03. Subsequently, 10 uniform outliers (i.e., 5% of the normal data) are introduced and uniformly distributed in the [0.5, 1] interval. Total number of the data vectors in this dataset is 2940 including the 5% outliers. Figure 5.5 illustrates data distribution of dataset of a single node.

The second 2-D synthetic dataset uses a combination of two different Gaussian distributions with different means for each node. Each training dataset consists of 1400 records generated from Gaussian distribution $N(\mu_1, \sigma_1)$ and each testing dataset consists of 1400 records with Gaussian distribution $N(\mu_2, \sigma_2)$, where μ_1 = [0.3 0.3], μ_2 = [0.45 0.45], $\sigma_1 = \sigma_2 = 0.03$, where 5% of the normal data is introduced as outliers and uniformly distributed in the [0.5, 1] interval. Total number of the data vectors in this dataset is 2940 including the 5% outliers. The reason of choosing this dataset is to analyze ability of our online techniques to learn new normal behavior in a non-stationary data. Figure 5.6 illustrates data distribution of dataset of a single node.

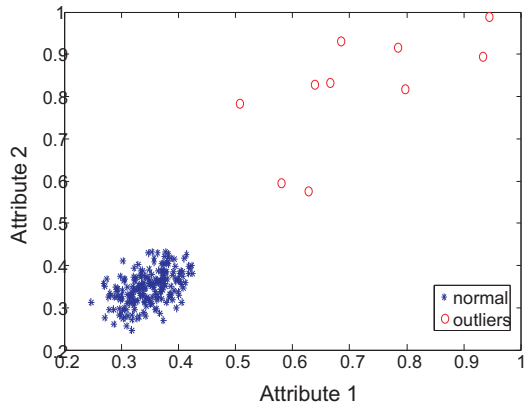


Figure 5.5: Data plot of a single node with mixed Gaussian distribution

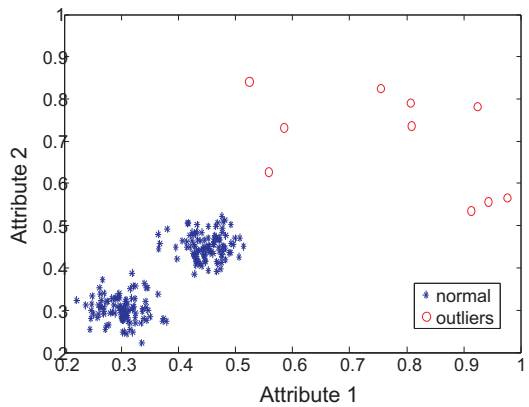


Figure 5.6: Data plot of a single node with varied distributions

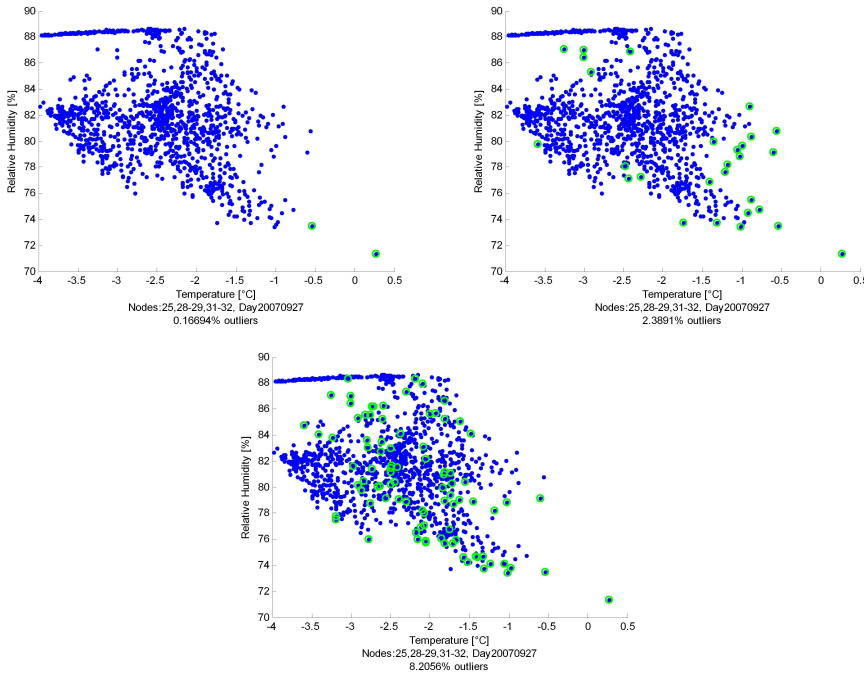


Figure 5.7: (left) Plot for labelled data based on Mahalanobis distance, (right) Plot for labelled data based on density, (lower) Plot for labelled data based on running average

For the real dataset, we use the dataset collected from nodes 25, 28, 29, 31, 32 at the Grand St. Bernard [108] for the period of 9am-17am on 27^{th} September 2007 with two attributes: ambient temperature and relative humidity. We label this dataset using different labelling techniques of Chapter 3, i.e., based on Mahalanobis distance, density and running average. Results of applying these labelling techniques are illustrated in Figure 5.7.

5.6.2 Experimental Results and Evaluation

For the simulation, we use Matlab [74] and test the following two non-linear kernel functions:

- Radial basis function (RBF): $k_{RBF} = exp(-\|x_1 - x_2\|^2/2\sigma^2)$, where σ is

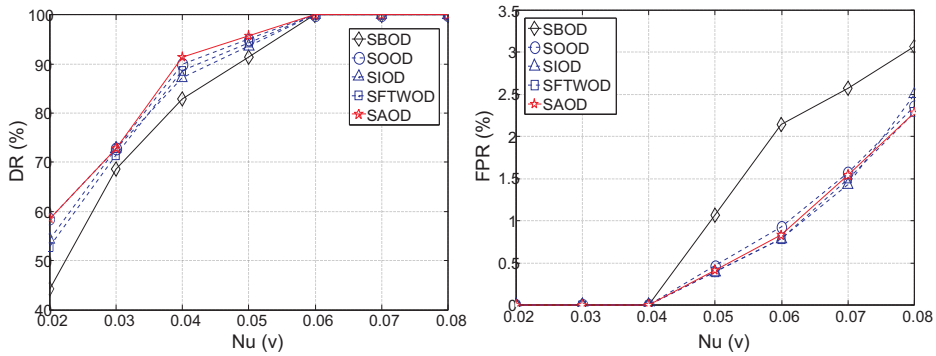
the width parameter of the kernel function;

- Polynomial function: $k_{Polynomial} = (x_1.x_2 + 1)^r$, where r is the degree of the polynomial function.

We use kernel matrices generated using the above kernel functions. In case of using RBF, kernel matrix should be centered. We evaluate two important performance metrics, the detection rate (DR), which represents the percentage of outliers that are correctly detected and the false alarm rate, also known as false positive rate (FPR), which represents the percentage of normal data that are incorrectly considered as outliers. DR represents the ratio between the number of correctly detected outliers and the total number of outliers, while FPR represents the ratio between the number of normal data detected as outliers and the total number of normal data.

We examine the effect of the regularization parameter υ for SOOD, SIOD, SFTWOD, SAOD and SBOD, in the feature space using the RBF and polynomial kernel functions as well as in the input space. As we described before, υ represents the fraction of data vectors that can be outliers. Larger parameter υ results in better detection rate, however, it can also lead to the higher false alarm rate. So an appropriate value for the parameter υ is exact ratio of outliers. In the case of having no a priori knowledge about the outliers ratio, variable υ can be used to evaluate the robustness of the techniques. This is due to the fact that a robust technique can achieve high accuracy rate while keeping a false alarm rate low regardless of increase or decrease of the parameter υ. In the experiments we have varied the parameter υ from 0.02 to 0.08 in intervals of 0.01. The kernel width parameter σ is set to 0.25, the kernel degree parameter r is set to 3, and the fixed size n used in SFTWOD is set to 10. A receiver operating characteristics (ROC) curve is usually used to represent the trade-off between the detection rate and the false alarm rate. The larger the area under the ROC curve, the better the performance of the technique.

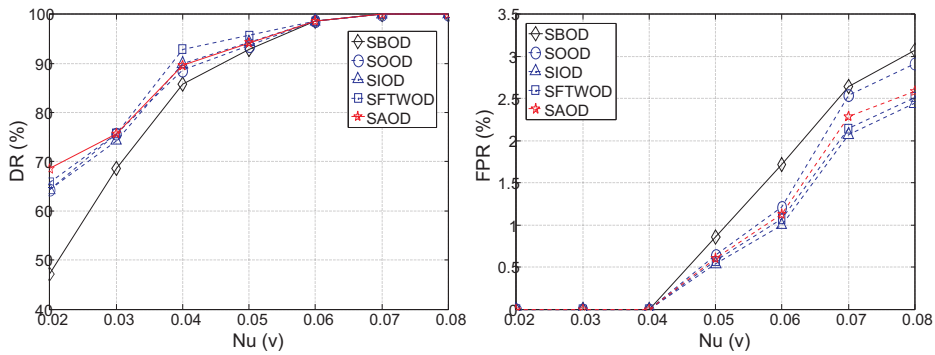
Figures 5.8, 5.9, and 5.10 show the detection rate and the false alarm rate obtained for our four online techniques as well as the SBOD offline technique in the feature space using the RBF and polynomial kernel functions as well as in the input space for the first synthetic data. We can see when the training and testing data are composed with same mixture Gaussian distribution, our four online techniques achieve better detection accuracy and lower false alarm compared with SBOD in presence of different υ parameter, as well as in both input space and feature space using different kernel functions. The good result of our techniques stem from using effective data preprocessing and the robustness of median and MAD. For the first synthetic dataset, SOOD presents high detection rate while ends up second in terms of high false alarm. On the other hands,



(a) Detection Rate Vs Nu (Input Space)

(b) False Alarm Rate Vs Nu (Input Space)

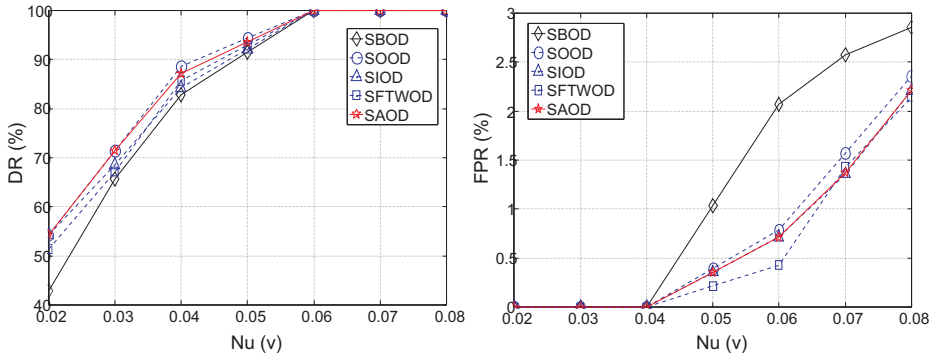
Figure 5.8: (a) Detection rate in the input space for synthetic data of mixture Gaussian distribution, (b) False alarm rate in the input space for synthetic data of mixture Gaussian distribution



(a) Detection Rate Vs Nu (RBF Kernel)

(b) False Alarm Rate Vs Nu (RBF Kernel)

Figure 5.9: (a) Detection rate with RBF kernel for synthetic data of mixture Gaussian distribution, (b) False alarm rate with RBF kernel for synthetic data of mixture Gaussian distribution



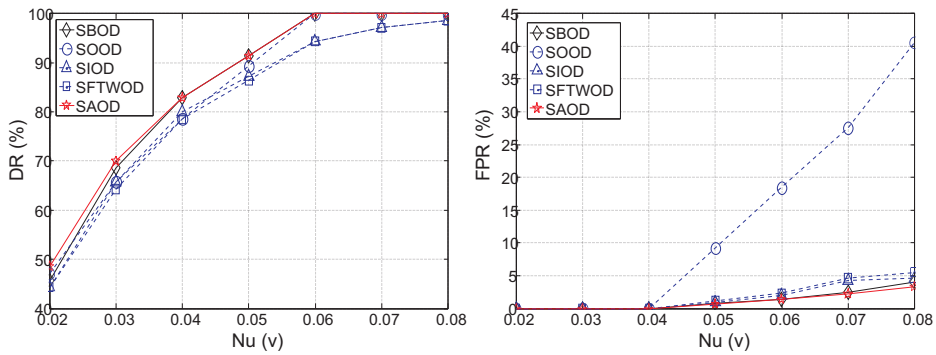
(a) Detection Rate Vs Nu (Polynomial Kernel)

(b) False Alarm Rate Vs Nu (Polynomial Kernel)

Figure 5.10: (a) Detection rate with polynomial kernel for synthetic data of mixture Gaussian distribution, (b) False alarm rate with polynomial kernel for synthetic data of mixture Gaussian distribution

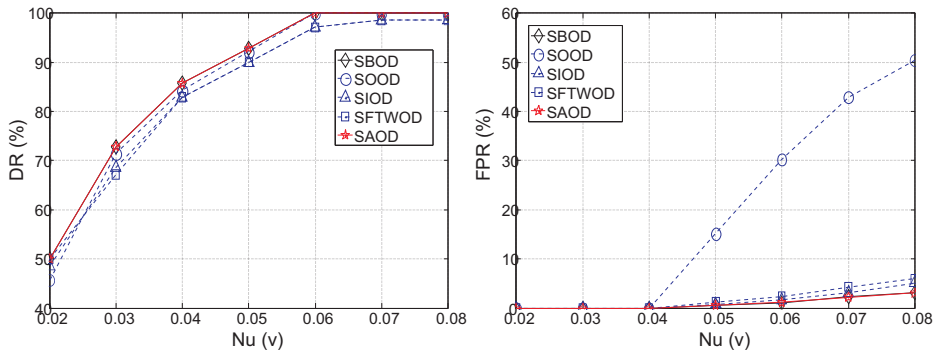
SAOD has high detection rate with a low false alarm and it also does not need to update the normal behavior at the end of time sequence due to the fact the faction of outliers is smaller than the upper bound (ν). SIOD and SFTWOD have similar performance in terms of detection rate and false alarm rate.

Figures 5.11, 5.12, and 5.13 show the detection rate and the false alarm rate obtained for our four online techniques as well as the SBOD offline technique using the RBF and polynomial kernel functions as well as in the input space for the second synthetic dataset. We can clearly see when the training and testing data are composed of different distributions, SOOD, SIOD and SFTWOD do not produce satisfactory results. Especially, SOOD causes very large false alarm. This is because SOOD does not update the model, while the data distribution of the dateset has changed. In contrary, SBOD obtains better detection rate and lower false alarm due to the fact that SBOD identifies all data points belonging to the new distribution as normal in an offline manner after all points are inserted. SAOD also achieves good performance although some data points were detected as outliers at the time of insertion. Initially, inserted data vectors from a new distribution are detected by SAOD as outliers. However, when the "switch" to the new distribution was complete, SAOD checks the number of detected outliers with the upper bound and then learns the new distribution as a part of regular behavior, thus the new data vector were correctly detected as an indication of normal behavior. In this case, the result of SAOD is still slightly better than SBOD since it has better understanding of data structure and alleviates the



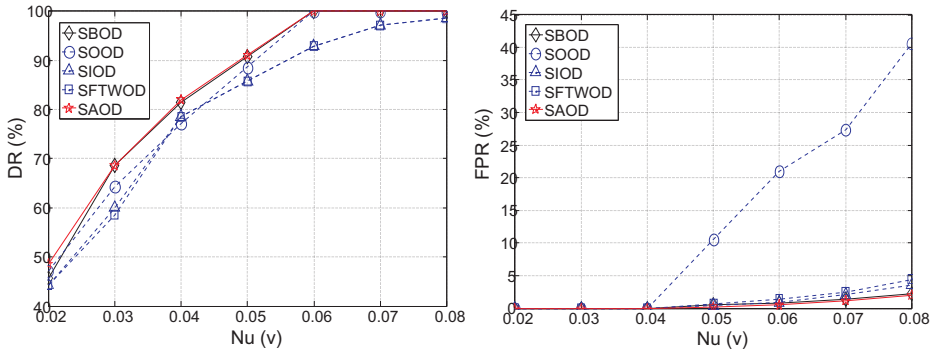
(a) Detection Rate Vs Nu (Input Space) (b) False Alarm Rate Vs Nu (Input Space)

Figure 5.11: (a) Detection rate in the input space for synthetic data of varied Gaussian distributions, (b) False alarm rate in the input space for synthetic data of varied Gaussian distributions



(a) Detection Rate Vs Nu (RBF Kernel) (b) False Alarm Rate Vs Nu (RBF Kernel)

Figure 5.12: (a) Detection rate with RBF kernel for synthetic data of varied Gaussian distributions, (b) False alarm rate with RBF kernel for synthetic data of varied Gaussian distributions



(a) Detection Rate Vs Nu (Polynomial Kernel) (b) False Alarm Rate Vs Nu (Polynomial Kernel)

Figure 5.13: (a) Detection rate with polynomial kernel for synthetic data of varied Gaussian distributions, (b) False alarm rate of polynomial kernel for synthetic data with varied Gaussian distributions

influence of outliers by using median and MAD. On the other hand, SBOD did not detect any change of normal behavior between two different distributions while SAOD did. SIOD and SFTWOD perform better than SOOD in terms of having lower false alarm rate. SIOD performs even better than SFTWOD as it cooperatively identifies outliers using spatial information at each time instant.

Figure 5.14 shows ROC curve and the false alarm rate obtained for our four online techniques as well as the SBOD offline technique only in the input space for the real dataset labelled by Mahalanobis distance. It can be seen that SOOD produces high false alarm due to the fact that it does not update the normal profile. Both SAOD and SBOD have achieved good performance while our SAOD has better accuracy than SBOD due to use of median and MAD parameters, as well as spatial information of neighboring nodes. The performance obtained by SIOD and SFTWOD are similar, while SIOD is slightly better than SFTWOD.

Figure 5.15 shows ROC curve obtained for our four online techniques as well as the SBOD offline technique in the input space for real dataset labelled by running average and density labelling techniques. All these five techniques show similar performance as the second synthetic data. Specifically, our SAOD's performance is better than SBOD using both labelling techniques. SOOD has the highest false alarm rate while keeping high detection rate. SIOD performs slightly better than SFTWOD. However, all techniques have not achieved good detection accuracy while generating high false alarm. The reason for this is that one-class SVM-based techniques essentially belong to distance-based techniques. This implies

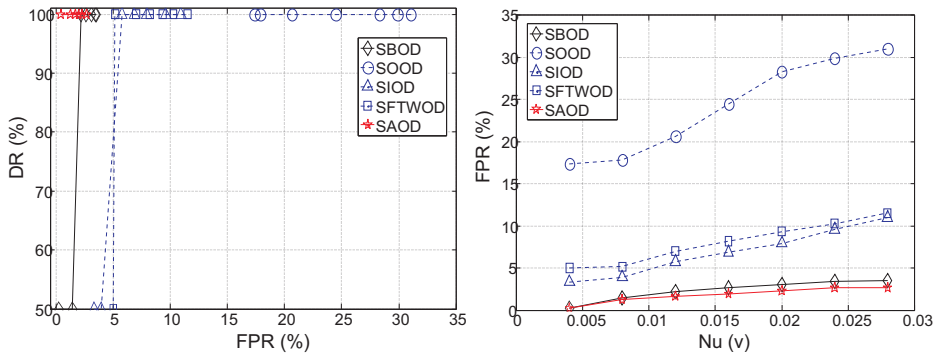


Figure 5.14: (a) ROC curve in the input space for labelled data based on Mahalanobis distance, (b) False alarm rate in the input space for labelled data based on Mahalanobis distance

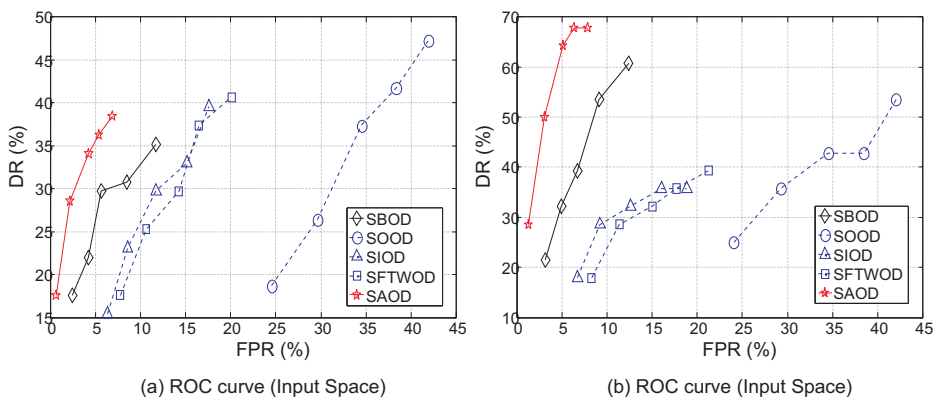


Figure 5.15: (a) ROC curve in the input space for labelled data based on running average, (b) ROC curve in the input space for labelled data based on density

that compared to density-based techniques, one-class SVM-based technique is not good at discovering local outliers, especially in datasets with diverse densities and arbitrary shapes. Running average labelling techniques combine the outliers identified from temperature and humidity individually (in 1D and not in 2D) and also identify outliers at a few small time intervals.

5.6.3 Complexity Analysis

We further compare these techniques in terms of communication overhead and computation and memory complexity. The communication complexity of our distributed techniques depends on the transmission of local quarter-sphere radius information as well as the median and MAD parameters. The maximum communication overhead in SOOD for each node is $O(d)$, where d is the dimension of observations. Each node only transmits its local radius information as well as the median and MAD once at the initial training phase. In SIOD, each node sends updated radius information as well as updated median and MAD at each time interval. The maximum communication overhead for each node in SIOD is $O(md)$, where m is the number of new observations to be classified. In SFTWOD, each node transmits updated radius information as well as the median and MAD at a fixed-size time interval. The maximum communication overhead of SFTWOD for each node is $O(\frac{m}{n}d)$, where n is the fixed size of time interval. SAOD requires no update of radius information during online outlier detection and only possibly communicates the updated median, MAD and radius information with nodes after a complete sliding time window. The maximum communication overhead of SAOD for each node is approximately equal to $O(d)$.

The computational complexity in SOOD is about computation of the median, the MAD, the linear optimization function and the distance between every new observation and the origin. The computational complexity of our techniques mainly depends on solving a linear optimization problem, which is represented as $O(p)$. Hence, the maximum computational complexity of each node in SOOD and SAOD is $O(mp)$, where m is the number of new observations to be classified, while for SIOD and SFTWOD the maximum computational complexity is $O(m^2p)$ and $O(\frac{m^2}{n}p)$, respectively. SBOD still needs to compute kernel matrix and the transformation of centered kernel function (especially for RBF kernel function), whose complexity is represented by $O(k)$. Thus the maximum computational complexity of SBOD for each node is $O(kmp)$.

The memory complexity of our techniques is mainly about keeping observations of the size of sliding window in memory being represented as $O(md)$, where d is the dimension of observations and m is the number of new observations to be classified. Overhead of storing other parameters such as statistical parameters

Techniques	Communication Complexity	Computational Complexity	Memory Complexity
SBOD	–	$O(kmp)$	$O(md + m^2)$
SOOD	$O(d)$	$O(mp)$	$O(md)$
SIOD	$O(md)$	$O(m^2p)$	$O(md)$
SFTWOD	$O(\frac{m}{n}d)$	$O(\frac{m^2}{n}p)$	$O(md)$
SAOD	$O(d)$	$O(mp)$	$O(md)$

Table 5.4: Complexity analysis of five outlier detection techniques for each sensor node

and the radius information is negligible. Hence the maximum memory complexity of each node for our techniques is $O(md)$. Due to the fact that SBOD need to keep $m \times m$ kernel function, its memory complexity of each node is $O(md + m^2)$.

Table 6.3 summarizes these complexity, from which we can see that SAOD has the lower computational and communication complexity compared with the other techniques. Moreover, SAOD can not only accurately and instantly detect outliers but also detect the change of normal behavior.

5.7 Chapter Summary

In this chapter we have proposed distributed and online outlier detection techniques based on quarter-sphere one-class SVM and theory of spatial and temporal correlations to precisely detect outliers. To cope with the problem of generating high false alarm rate, we also propose three updating strategies to incorporate new arrived observations and update the modelled quarter-sphere SVM for more reliable outlier detection and detect the change of normal behavior of sensor data. We compare performance of our four techniques with a previously proposed batch technique using both synthetic and real datasets as well as different labelling techniques. Experimental results show that our SAOD has the ability to accurately detect outliers and the change of normal behavior in sensor data streams. It is also robust in terms of parameter selection, while keeping the communication, computational complexity and memory costs low.

Chapter 6

Ellipsoidal SVM-Based Outlier Detection Techniques for Wireless Sensor Networks

Sensor data attributes are often correlated so that they are not always distributed around the center of mass in a perfect spherical shape. Compared with spherical models, an ellipsoidal model takes into account the correlation between data attributes and more precisely capture the multivariate data structure. In this chapter, we introduce a hyperellipsoidal one-class SVM and use it to propose our distributed and online outlier detection techniques to identify multivariate outliers in WSNs. We also take advantage of the theory of spatio-temporal correlations to identify outliers and update the SVM-based model representing normal behavior of sensor data for further outlier identification. Experimental results show that our ellipsoidal SVM-based outlier detection techniques achieve better detection accuracy and lower false alarm, as compared to previously proposed spherical SVM-based outlier detection techniques.

6.1 Introduction

In Chapter 5, we have described how to efficiently model the quarter-sphere one-class SVM classifier in WSNs and utilized this classifier to design our distributed and online outlier detection techniques for WSNs. These quarter-sphere SVM-based techniques require no a priori knowledge about data distribution or labelled data for training and effectively identify outliers in multivariate sensor data. They also detect and adapt to changes of normal behavior of sensor data.

It can be seen that modelling the quarter-sphere SVM classifier in essence is a process of determining similarity among given data vectors. By using certain *distance measure*, the majority of data vectors with high similarity are enclosed within a quartersphere, which represents the normal behavior of given data vectors. This manner of encompassing data vectors into the quartersphere assumes that the data vectors are distributed around the center of mass in a perfect *sphere* [132]. It considers the data attributes independent of each other and assumes that there is no inherent correlation between them. However, in reality sensor data attributes are often *correlated*, e.g., as mentioned in Chapter 3 temperature has certain correlation with humidity. Consequence of ignoring the correlation between data attributes in sphere-like models is that the modelled normal boundary is not sufficiently precise to represent the normal behavior of data vectors and it is very likely to impact the results of outlier detection by high generation of false alarms for multivariate sensor data. On the other hand, they may miss real *multivariate outliers*, in which none of their attributes are outliers. An example of this situation is an observation with relatively high temperature and humidity, while temperature and humidity are negatively correlated.

A good way for analyzing multivariate data is to take into account the existing correlation between data attributes and use specific distance measure to determine the similarity of data vectors. In this way, the data vectors are not distributed around the center of mass in a spherical manner, instead their distribution will have a *ellipsoidal* shape [64]. The direction of the formed ellipsoid reveals the multivariate data distribution trend, as well as the strength of the correlation between data attributes. This ellipsoid modelling enables to precisely capture multivariate data structures by considering not only the distance from the center of mass but also data attribute correlation. It, therefore, alleviates the problems caused by assuming spherical shapes while detecting outliers in multivariate data.

In this chapter, we use the *ellipsoidal one-class SVM* [132] approach to identify multivariate outliers in resource-constrained WSNs. Our proposed outlier detection techniques are designed to operate in a distributed and online manner. In the process of detecting outliers, we also take advantage of the theory of spatio-

temporal correlations to precisely detect outliers and changes of normal behavior of sensor data. Furthermore, we present an adaptive technique to update the ellipsoidal SVM-based model that represents changing normal behavior of sensor data. Experiments with two synthetic datasets and real environmental dataset from the Grand St. Bernard [108] show that our proposed outlier detection techniques achieve better detection accuracy and lower false alarm as compared to our quarter-sphere SVM-based technique proposed in Chapter 4 and existing offline SVM-based outlier detection techniques [99, 100] designed for WSNs.

The remainder of this chapter is organized as follows. Related work on using ellipsoidal one-class SVM-based outlier detection in WSNs as well as in data mining and machine learning communities is described in Section 6.2. Principles of modelling the ellipsoidal one-class SVM classifier are addressed in Section 6.3. How to lower down the complexity of traditional hyperellipsoidal one-class SVM classifier to fit requirements of WSNs is addressed in Section 6.4. Our proposed ellipsoidal SVM-based outlier detection techniques are presented in Section 6.5. Experimental results and performance evaluation of our techniques are reported in Section 6.6. Finally this chapter is concluded in Section 6.7.

6.2 Related Work

A straightforward method of outlier detection is to find a boundary that appropriately encloses the available observations such that the chance of misclassification of unseen observations can be minimized. Compared with supervised techniques, unsupervised techniques used in data mining and machine learning communities typically do not require labelled data to learn normal and abnormal models and detect outliers as data instances that are significantly different from a shape-based normal boundary, which encompasses the majority of data points depending on similarity measure [10]. Based on the assumption that number of anomalous data in a dataset is considerably smaller than number of normal data, they only model normal behavior of the unlabelled data while automatically ignoring the anomalies existing in the training set.

SVMs have received great attention as efficient non-parametric classification and regression tools in the data mining and machine learning communities [106]. They can separate the data belonging to different classes by fitting a hyperplane among them which maximizes the separation. The one-class SVM constitutes the extension of the main SVM ideas from supervised to unsupervised paradigms [68]. This method permits the control of the number of outliers in the training set and the solution of the optimization problem leads to a decision function which classifies new observations as normal or outliers. In addition to the two typical

one-class SVM-based techniques [106, 122] utilizing shape-based normal boundary addressed in Chapter 5, Wang et al. [132] have proposed a *hyperellipsoid-based* one-class SVM, which identifies outliers by fitting multiple hyperellipsoids with *minimum effective radii*. The data vectors falling outside the hyperellipsoids are declared as outliers. Campbell et al. [23] and Laskov et al. [68] have proposed the linear optimization solution to reduce high computational complexity of modelling one-class SVM classifiers.

Rajasegarar et al. [100] have further extended work in [132, 68] by proposing a hyperellipsoidal one-class SVM with the linear optimization problem. However, this technique is neither distributed nor online. It only operates in an independent location in an offline manner and therefore is not suitable for many WSN applications.

In this chapter, we propose distributed and online outlier detection techniques based on simplified hyperellipsoidal one-class SVM with the linear optimization problem. Our techniques enable to identify outliers in an online manner and detect and adapt to changes of normal behavior of sensor data in real-time.

6.3 Principles of Modelling Hyper-Ellipsoid One-Class SVM

In this section, we describe the principles of modelling the hyperellipsoidal one-class SVM proposed in [132] for multivariate data vectors. The quadric optimization problem of modeling the hyperellipsoidal SVM classifier has been converted to the linear optimization problem in [100] by fixing the center of mapped data vectors in the feature space at the origin. The geometry of hyperellipsoidal one-class SVM-based approach is shown in Figure 6.1. The general process of modeling the hyperellipsoidal SVM classifier for multivariate data vectors is addressed below.

Assume that m data vectors $\{x_i \epsilon \Re^d, i = 1, \ldots, m\}$ of d variables in the input space are mapped into the feature space using some non-linear mapping function ϕ. The hyperellipsoidal SVM aims at enclosing a majority of mapped data vectors $\phi(x_i)$ in the feature space by fitting a hyperellipsoid centered at the origin with a minimum *effective radius* R. Thus, the optimization problem in this hyperellipsoidal SVM classifier is represented as:

$$\min_{R \epsilon \Re, \xi \epsilon \Re^m} \quad R^2 + \frac{1}{vm} \sum_{i=1}^{m} \xi_i \tag{6.1}$$

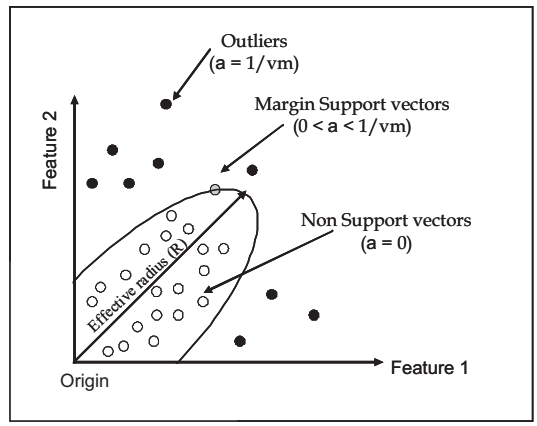


Figure 6.1: Geometry of the hyper-ellipsoidal formulation of one-class SVM [100]

$$subject\ to : \phi(x_i)\Sigma^{-1}\phi(x_i)^T \leq R^2 + \xi_i, \xi_i \geq 0, i = 1, 2, \ldots m$$

where $\upsilon\ \epsilon\ (0, 1)$ is a parameter that controls the fraction of mapped data vectors that can be outliers, and slack variables $\{\xi_i : i = 1, 2, \ldots m\}$ are denoted to allow some of mapped data vectors to lay outside the hyperellipsoid. Σ^{-1} is the inverse of the covariance matrix Σ of mapped data vectors, which is computed as follows:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(\phi(x_i) - \mu)(\phi(x_i) - \mu)^T, \quad \mu = \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) \tag{6.2}$$

Using Mercer Kernels [128], the inner products of mapped data vectors in the feature space can be computed in the input data space without needing any knowledge about the non-linear function ϕ. Let $K\epsilon\Re^{m\times m}$ be the kernel matrix of the original data vectors. Similarly, mapped data vectors can be centered in the feature space by subtracting the mean. Then the centered kernel matrix K_c can be obtained in terms of the kernel matrix K using $K_c = K - 1_m K - K 1_m + 1_m K 1_m$, where 1_m is the $m \times m$ matrix with all its values equal to $\frac{1}{m}$.

The eigen structures of K_c is denoted by $K_c = A\Omega A^T$, where Ω is a diagonal matrix with positive eigenvalues as the diagonal elements, and A is the eigenvector matrix corresponding to the positive eigenvalues [40]. Hence the covariance matrix Σ can be denoted as $\Sigma = (\Omega^{-\frac{1}{2}}A\phi(x)^T)(\frac{\Omega}{m})(\Omega^{-\frac{1}{2}}A\phi(x)^T))^T$, where X is the data vectors in feature space. By calculating the pseudo inverse Σ^+, we

can approximate Σ^{-1} as $\Sigma^{-1} = \Sigma^{+} = mX^T A\Omega^{-2}A^T X$ [132]. Consequently, Equation 6.1 will become as follows:

$$\min_{R\epsilon\Re,\xi\epsilon\Re^m} \quad R^2 + \tfrac{1}{\upsilon m}\sum_{i=1}^{m}\xi_i \tag{6.3}$$

$$subject\ to: \|\sqrt{m}\Omega^{-1}A^T K_c^i\|^2 \leq R^2 + \xi_i, \xi_i \geq 0, i = 1, 2, \ldots m$$

where K_c^i is the i^{th} column of the kernel matrix K_c. Using similar Lagrange function and deviations as explained in Chapter 5, finally the dual formulation of hyper-ellipsoidal SVM will become a linear optimization problem represented as:

$$\min_{\alpha\epsilon\Re^m} \quad -\sum_{i=1}^{m}\alpha_i\|\sqrt{m}\Omega^{-1}A^T K_c^i\|^2 \tag{6.4}$$

$$subject\ to: \sum_{i=1}^{m}\alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{\upsilon m}, i = 1, 2, \ldots m$$

The data vectors with $\alpha_i = 0$ will fall inside the hyperellipsoid and will be considered as normal. The data vectors with $0 < \alpha_i < \frac{1}{\upsilon m}$ will reside on the surface of the hyperellipsoid. Their distances to the hyperellipsoidal center indicate the minimum effective radius R, which can be obtained by calculating $R^2 = \|\sqrt{m}\Omega^{-1}A^T K_c^i\|^2$ for any margin support vectors. Those data vectors with $\alpha = \frac{1}{\upsilon m}$ whose distances to the origin are larger than R of the hyperellipsoid are considered as outliers.

6.3.1 Hyper-Ellipsoid SVM VS. Hyper-Sphere SVM

We have described hyper-ellipsoid SVM and hyper-sphere SVM in this chapter and Chapter 5, respectively. Both hyper-ellipsoid SVM and hyper-sphere SVM are used to model the normal behavior of given data vectors. A significant difference between the two SVMs is that they use different distance measures to determine the similarity of data vectors and further model the normal behavior of data vectors. More specifically, hyper-sphere SVM uses *Euclidean distance* (ED) while hyper-ellipsoid SVM uses *Mahalanobis distance* (MD). The two distance measures are both commonly used to measure the similarity between any two data vectors [118]. Euclidean distance does not consider the correlation between attributes and calculates the distance in terms of individual attribute.

On the contrary, Mahalanobis distance considers the correlation between attributes and calculates the distance by combining all attributes together. This correlation between attributes can be represented by *covariance matrix*, where variance of a variable itself and covariance between any two variables are included. Formally, given multivariate data vectors $x = (x_1, x_2, x_3, ..., x_N)$ with mean $\mu = (\mu_1, \mu_2, \mu_3, ..., \mu_N)$ and covariance matrix Σ, the Mahalanobis distance of these data vectors is defined as:

$$MD(x) = \sqrt{(x-\mu)\Sigma^{-1}(x-\mu)^T} \tag{6.5}$$

If the covariance matrix Σ is the identity matrix I, where all diagonal elements are set to 1, Mahalanobis distance reduces to the Euclidean distance for two data vectors x and y and is represented as:

$$EM(x,y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{6.6}$$

Compared to Euclidean distance, Mahalanobis distance, which takes into account both distance from the center of mass and the direction, has a better understanding of multivariate data structure as Euclidean distance is blind to attribute correlation and assumes all data vectors have equal distance from the center of mass. Moreover, Mahalanobis distance is *scale-invariant* meaning that it is independent on the scale of data attributes, while Euclidean distance is extremely sensitive to the scale of data attributes. However, the computational and memory complexity of Mahalanobis distance is much higher than Euclidean distance due to computation of covariance matrix. Euclidean distance is simple to calculate. Table 6.1 clearly shows the comparison between the two distance measures.

Consequently, using Euclidean distance for as similarity measure would generate a sphere at 2-D data space, where data vectors are equally distributed around the center of mass. Using Mahalanobis distance as similarity measure would generate an ellipse at 2-D data space, where data vectors are distributed in directional linear trend [45], which indicates the correlation between variables. The two different shapes actually define the normal behavior of data vectors. We here introduce an example to represent the results of outliers using the two shapes. Figure 6.2 illustrate the normal behaviors of data vectors modelled by hyper-ellipsoid SVM and hyper-sphere SVM using corresponding distance measures, respectively. As seen from Figure 6.2, outliers detected by the sphere may not be considered as outliers by the ellipse (*point B*); whereas, data vectors that are not declared as outliers may be considered as outliers by the ellipse (*point A*).

Classifiers	Distance measure	Characteristics	Shape
Hyper-ellipsoid SVM	Mahalanobis distance	- Considers attribute correlation - Scale-invariant - High complexity	Ellipse
Hyper-sphere SVM	Euclidean distance	- Ignores attribute correlation - Scale-sensitive - Low complexity	Sphere

Table 6.1: Comparison between hyper-ellipsoid SVM and hyper-sphere SVM

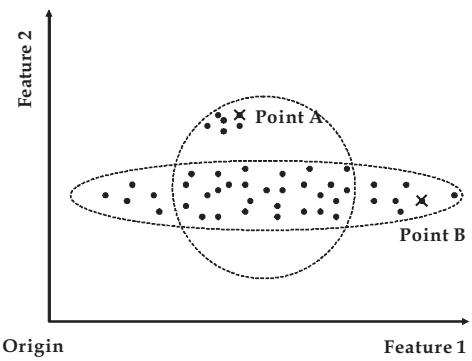


Figure 6.2: Similarity measure between Mahalanobis distance and Euclidean distance. Using Mahalanobis distance generates an ellipse while using Euclidean distance generates a sphere at 2-D space.

Therefore, using an appropriate shape and its corresponding distance measure to model normal behavior of data vectors is significantly important for accurate outlier detection. The choice between Euclidean distance and Mahalanobis distance depends on data characteristics and requirements of applications.

In this chapter, we use hyper-ellipsoid SVM to model the normal behavior of sensor data, which is based on the fact that sensor data are often correlated as also shown by experimental results in Chapter 3. We would further reduce the computational complexity of modelling hyper-ellipsoid SVM for resource-constraint WSNs.

6.4 Fitting Hyper-Ellipsoid One-Class SVM Modelling to Resource-Constraint WSNs

We have compared hyper-ellipsoid SVM and hyper-sphere SVM regarding modeling the normal behavior of data vectors and identifying outliers. We are aware that modelling hyper-ellipsoid SVM has high computational and memory complexity due to the fact that it considers the attribute correlation and also it generates kernel matrix and has the transformation of central kernel matrix. To reduce resource cost of modelling hyper-ellipsoid SVM in the feature space, we model hyper-ellipsoid SVM in the *input space* and fix the center of hyperellipsoid at the origin. For doing so, raw sensor data may still need to be transformed to a better symmetric data distribution using Box-Cox method. Then due to the fact that Mahalanobis distance is scale-invariant, data vectors can be centered at the origin only by subtracting the mean. For a data vector x_i, its *mean-centered value* is formulated as $x_i^{'} = (x_i - \mu)$. Considering that mean-centered values may be sensitive to outliers, we replace the arithmetic mean by the *median*.

After data preprocessing, the data vectors are centered at the origin in the input space to lower down the computational and memory complexity of modelling hyper-ellipsoid SVM in the feature space. Consequently, the dual formulation of Equation 6.4 in the input space will be simplified as:

$$\min_{\alpha \epsilon \Re^m} \quad -\sum_{i=1}^{m} \alpha_i (x_i^{'} \Sigma^{-1} x_i^{'T}) \tag{6.7}$$

$$subject\ to: \sum_{i=1}^{m} \alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{vm}, i = 1, 2, \ldots m$$

where $x_i^{'}\Sigma^{-1}x_i^{'T}$ represents the Mahalanobis distances of mean-centered data vectors in the input space from the origin. We further present a basic decision function to determine whether a new arriving sensor observation x is an outlier or not using the modelled hyper-ellipsoid SVM in the input space. According to Equation 6.7, the decision function can be computed as:

$$f(x) = sgn(R^2 - d(x^{'})^2) = sgn(R^2 - x^{'}\Sigma^{-1}x^{'T}) = sgn(R^2 - \|\Sigma^{-\frac{1}{2}}x^{'T}\|) \quad (6.8)$$

where R^2 is the square of the effective radius of the hyperellipsoid and it can be computed by the inner product of any margin support vectors in the input space. The observations with a negative value are classified as outliers since their square distances from the origin depending on the direction in the input space are larger than R^2. It can be obviously seen from Equation 6.4 that computation of the decision function in the input space is cheaper than in the feature space. Also, modelling hyper-ellipsoid SVM in the input space solves the problem of impossible calculation of inverse matrix of Σ when Σ is singular [129].

6.5 Ellipsoidal SVM-Based Outlier Detection Techniques

In this section, we use the modelled hyper-ellipsoid SVM to identify multivariate outliers and detect changes of normal behavior of sensor data in an online manner. We consider the same network topology illustrated in Figure 4.6 in Chapter 4.

We use the similar strategy as of SOOD for our ellipsoidal SVM-based outlier detection technique. Our ellipsoidal SVM-based outlier detection technique (EOOD) enables each node to determine its every new observation as normal or outlier in real-time, according to Equation 6.8. The transferred parameters become the median, the covariance matrix and the effective radius of the hyper-ellipsoid SVM modeled by m observations. Based on temporal correlation of sensor data, each node uses the hyper-ellipsoid SVM modeled at a short time window in the previous day to determine whether its new observations is normal or outliers at the corresponding time window in the next day. Moreover, each node communicates these parameters with its spatially neighboring nodes to cooperatively identify outliers based on spatial correlation of sensor data. The main steps of EOOD are:

- *Step 1.* Each sensor node s_i models its hyper-ellipsoid SVM for m observations and then obtains the threshold R_i as well as the corresponding parameters of the median and the covariance matrix of centered data. Then

the local outliers can be determined in real-time by comparing the distances between new observations and the origin with R_i using Equation 6.8.

- *Step 2.* Each node s_i communicates its parameters of R_i as well as the median and the covariance matrix of its spatial neighbors. Due to the fact that the covariance matrix is symmetric, each node only transmits *d(d+1)/2* elements for covariance matrix, where d is the dimension of a sensor observation.

- *Step 3.* Each node s_i collects these parameters of R_i as well as the median and the covariance matrix from its spatial neighbors. Then it combines these parameters together with its own R_i, median and covariance matrix (covariance matrix can be merged by [64]). The merged parameters are denoted as R_g, the global median, and the global covariance matrix.

- *Step 4.* Each node s_i uses its merged parameters to online determine global outliers for its new observations. For a new observation x, the decision function indicating whether it is a global outlier in the input space can be defined as:

$$f(x) = sgn(R_g^2 - \|\Sigma^{-\frac{1}{2}} x^{'T}\|) \tag{6.9}$$

EOOD scales well with increase of number of nodes due to its distributed processing nature. It can update the merged parameters by communicating among spatially neighboring nodes at the end of each time interval. It lowers down the communication overhead and computational complexity and does not need to transmit any actual observations between sensor nodes except the parameters. EOOD can also be extended to identify various types of outliers in real-time using the same strategies described in Chapter 4 and detect the change of normal behavior occurred in two consecutive time windows.

EOOD does not update the existing SVM model until the end of the entire time interval. It indeed can detect the change of normal behavior between two consecutive time windows when most of observations measured in the next time window are detected as outliers. However, it can not detect changes in normal behavior of data within the time window or adapt to new behavior of sensor data. Therefore, EOOD may suffer from a possibly high rate of false alarm since new normal observations are detected as outliers. In order to alleviate this problem, we use the same strategy of SAOD to incorporate new arrived data and sequently update the normal model depending on previous decision results for future outlier detection.

6.5.1 Ellipsoidal SVM-Based Adaptive Outlier Detection Technique (EAOD)

Here we introduce an adaptive outlier detection technique (EAOD), based on the relationship between the previous decision results and the modeled hyper-ellipsoid SVM. We use the same size (m) as of sliding window to model hyper-ellipsoid SVM for normal behavior of sensor data. When all new observations are instantly detected as normal or outlier and inserted in the entire sliding window, each node will check if the number of detected outliers exceeds the given upper bounder (ν). If so, a new normal behavior is detected and the SVM model should be updated. After updating the hyper-ellipsoid SVM, each observation can be labeled as normal or outlier according to Equation 6.9. EAOD reduces communication overhead and ensures the reliability of outlier detection results. The corresponding pseudocode for EAOD is shown in Table 6.2.

6.6 Experiments

This section describes performance evaluation of our EOOD and EAOD, compared to SAOD proposed in Chapter 5, and EBOD and SBOD presented earlier in Rajasegarar et al. [99, 100]. The goals of our evaluation are (i) to test the accuracy of our distributed and online outlier detection techniques and their robustness in terms of parameter selection, (ii) to compare the performance between ellipsoidal and spherical SVM-based techniques, and (iii) to investigate impact of different labelling techniques described in Chapter 3 on performance of outlier detection techniques.

6.6.1 Experimental Datasets

In our experiments, we use two synthetic datasets as well as a real dataset gathered from the Grand St. Bernard [108]. For the simulation, we use Matlab and consider the same sensor sub-network in Chapter 5. We use the first 2-D synthetic dataset in Chapter 5 and also create a new synthetic dataset by changing the mean of a mixture of three Gaussian distributions into (0.25, 0.35, 0.45). The standard deviation is still 0.03 and 5% (of the normal data) anomalous data is introduced and uniformly distributed in the [0.5, 1] interval. Figure 6.3 illustrates different data distributions of the two synthetic datasets of a single node, respectively. The two synthetic datasets aim to evaluate the performance of ellipsoidal and spherical SVM-based outlier detection techniques for different data distributions.

```
1 procedure ModellingSVMProcess()
2   each node models the hyper-ellipsoid SVM;
3   each node locally broadcasts the modeled hyper-ellipsoid R_i as well as the median
    and the covariance matrix to its spatially neighboring nodes;
4   each node then computes the global R_g as well as the global median
    and covariance matrix;
5   initiate OutlierDetectionProcess(R_g, the global median and covariance matrix);
6   return;
7 procedure OutlierDetectionProcess(R_g, the global median and covariance matrix)
8   when x(t) arrives
9       compute d(x);
10      if (d(x) > R_g)
11          x(t) indicates an outlier;
12      else
13          x(t) indicates a normal observation;
14      endif;
15      initiate UpdatingSVMProcess(x(t));
16      set t ← t + 1;
17      if (m data observations are collected)
18        if (the number of detected outliers > the given upper bound (ν))
19          update the SVM model for (x(t − m + 1) ... x(t)) for outlier detection;
20        endif;
21      endif;
22  return;
23 procedure UpdatingSVMProcess(x(t))
24   update the sliding window: the oldest observations x(t − m)
     is removed and replaced by x(t);
25   update the median and the covariance matrix for the sliding window;
26   return;
```

Table 6.2: Pseudocode of EAOD

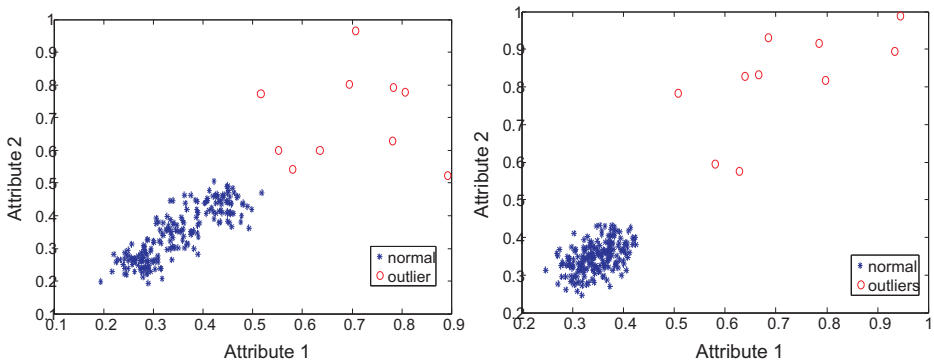


Figure 6.3: (left) Plot for a single node with ellipsoidal data distribution, (right) Plot for a single node with spherical data distribution

The real data still is collected from a cluster of neighboring sensor nodes, i.e., nodes 25, 28, 29, 31, 32 from the Grand St. Bernard [108]. In our experiments, we test the real data during the period of 6am-14am on 1^{st} October 2007 with two attributes: ambient temperature and relative humidity for each sensor observation. We label this dataset using different labelling techniques of Chapter 3, i.e., based on Mahalanobis distance, density and running average. Results of applying these labelling techniques are illustrated in Figure 6.4.

6.6.2 Experimental Results and Evaluation

We evaluate two important performance metrics, the detection rate (DR) and the false positive rate (FPR). We also examine the effect of the regularization parameter υ for EBOD, SBOD, EOOD, EAOD and SAOD in the input space. In the experiments we have varied υ between 0.02 and 0.08 in intervals of 0.01. A ROC curve is usually used to represent the trade-off between the detection rate and the false alarm rate.

Figures 6.5 shows the detection rate and the false alarm rate obtained for our online techniques EOOD, EAOD, SAOD as well as EBOD and SBOD offline techniques in the input space for the first synthetic data with ellipsoidal data distribution. We can see when data vectors are composed with ellipsoidal data distribution, our ellipsoidal SVM-based techniques EOOD and EAOD achieve better detection accuracy and lower false alarm compared with spherical SVM-based SBOD and SAOD in presence of different υ parameters. The good results of EOOD and EAOD stem from taking into account the correlation of data at-

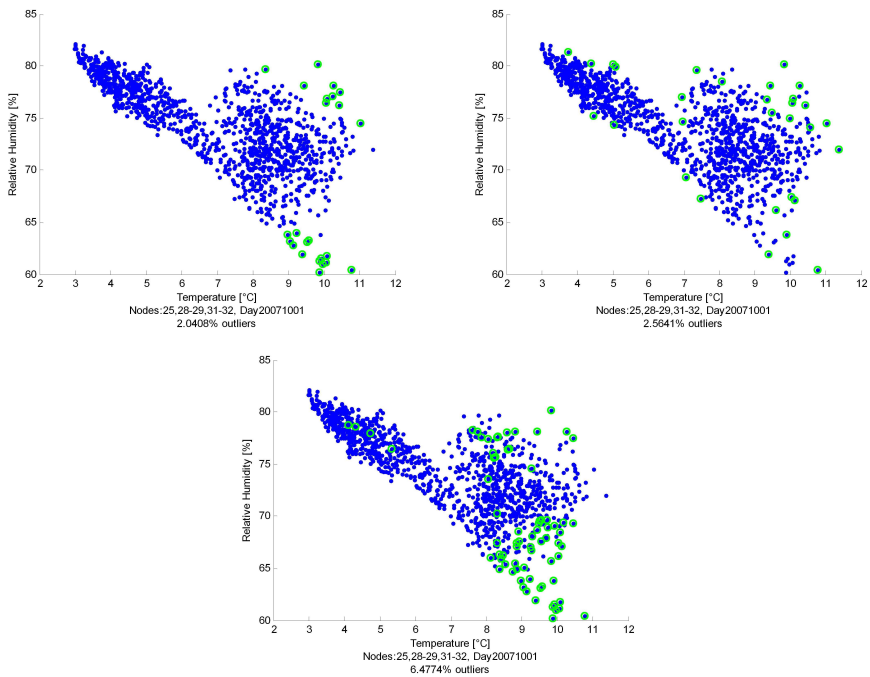
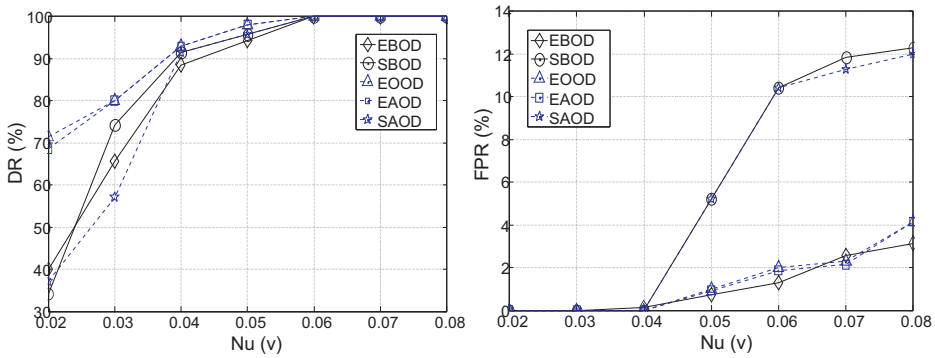


Figure 6.4: (left) Plot for labelled data based on Mahalanobis distance, (right) Plot for labelled data based on density, (lower) Plot for labelled data based on running average



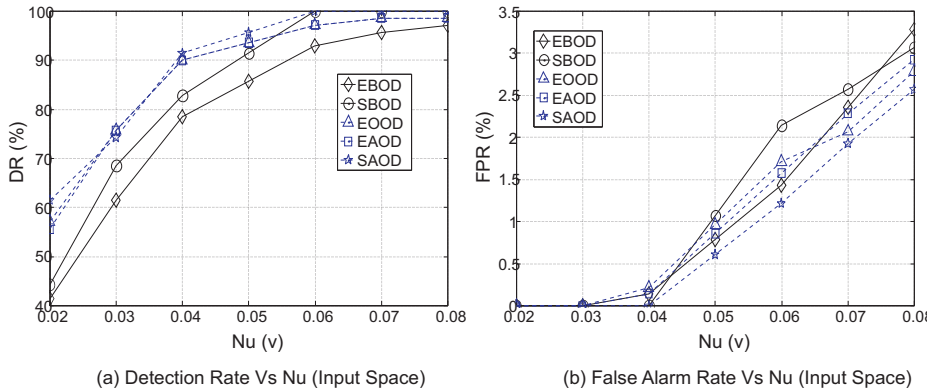
(a) Detection Rate Vs Nu (Input Space)

(b) False Alarm Rate Vs Nu (Input Space)

Figure 6.5: (a) Detection rate in the input space for synthetic data of ellipsoidal data distribution, (b) False alarm rate in the input space for synthetic data of ellipsoidal data distribution

tributes and having better understanding of multivariate data distribution. On the contrary, spherical SVM-based techniques ignore correlation between data attributes and use a spherical boundary to fit the data. This results in low detection rate and high false alarm rate in case of non-spherical data distribution. Furthermore, our EOOD and EAOD perform better than EBOD due to the fact that EOOD and EAOD exchange ellipsoidal information (median, covariance matrix) with spatial nodes for reliable outlier detection. On the other hand, EAOD does not need to update the normal behavior at the end of time sequence due to the fact the faction of outliers is smaller than the upper bound (ν).

Figures 6.6 shows the detection rate and the false alarm rate obtained for our online techniques EOOD, EAOD, SAOD as well as EBOD and SBOD offline techniques in the input space for the second synthetic data with spherical data distribution. We can clearly see when data vectors are composed of spherical data distribution, our spherical SVM-based technique SAOD achieves better detection accuracy and lower false alarm compared with ellipsoidal SVM-based EOOD, EAOD and EBOD. This is because spherical SVM-based techniques assume that data vectors are distributed around the center of mass in an ideal spherical shape. Although ellipsoidal SVM-based techniques take into account correlation of data attribute, they do not perform as good as spherical SVM-based techniques for spherical data distribution. Moreover, our SAOD is better than SBOD since SAOD alleviates the influence of outliers by using median and MAD and also exchanges these spherical information with spatial nodes for outlier detection.



(a) Detection Rate Vs Nu (Input Space)

(b) False Alarm Rate Vs Nu (Input Space)

Figure 6.6: (a) Detection rate in the input space for synthetic data of spherical data distribution, (b) False alarm rate in the input space for synthetic data of spherical data distribution

Furthermore, our EOOD and EAOD have better detection accuracy than EBOD.

Figure 6.7 shows ROC curve and the detection rate obtained for our online techniques EOOD, EAOD, SAOD as well as EBOD and SBOD offline technique in the input space for the real dataset labelled by Mahalanobis distance. It can be seen that EOOD produces the highest detection rate and highest false alarm due to the fact that it does not update the normal profile, while the data distribution has changed. EAOD has achieved the best performance with the highest detection accuracy and the lowest false alarm since it considers correlation of data attributes as well as use of ellipsoidal information (median, covariance matrix) from spatial nodes. Moreover, our SAOD is better than SBOD although both techniques do not produce satisfactory results due to ignoring attribute correlation of sensor data.

Figure 6.8 shows ROC curve obtained for our online techniques EOOD, EAOD, SAOD as well as EBOD and SBOD offline techniques in the input space for real dataset labelled by running average and density labelling techniques. The results show that our EAOD's performance is better than other techniques using both labelling techniques although it has not achieved good detection accuracy. EOOD still has the highest false alarm rate while keeping high detection rate.

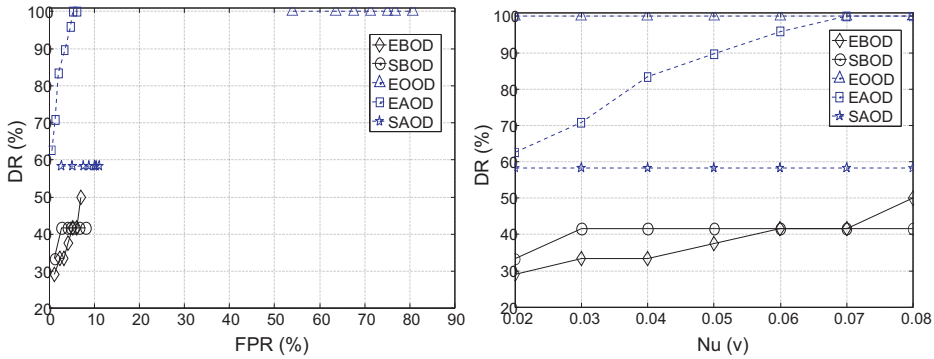


Figure 6.7: (a) ROC curve in the input space for labelled data based on Mahalanobis distance, (b) False alarm rate in the input space for labelled data based on Mahalanobis distance

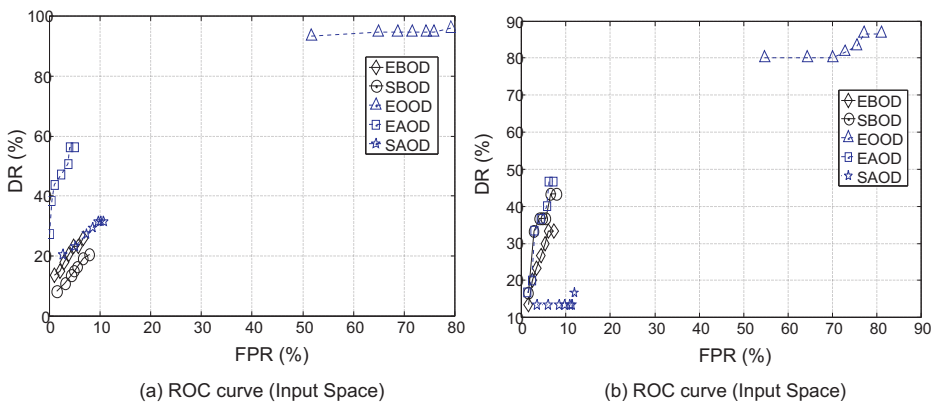


Figure 6.8: (a) ROC curve in the input space for labelled data based on running average, (b) ROC curve in the input space for labelled data based on density

6.6.3 Complexity Analysis

We further compare these techniques in terms of communication overhead and computation and memory complexity. The communication complexity of our distributed techniques depends on the transmission of local hyper-ellipsoid radius information as well as the median and covariance matrix parameters.

The communication overhead in EOOD for each node is $O(d^2)$, where d is the dimension of observations. Each node only transmits its local hyper-ellipsoid radius information as well as the median and covariance matrix once at the initial training phase. EAOD requires no update of radius information during online outlier detection and only possibly communicates the updated median, covariance matrix and radius information with nodes after a complete sliding time window. The maximum communication overhead of EAOD for each node is approximately equal to $O(d^2)$.

The computational complexity in EOOD is about computation of the median, the covariance matrix, the linear optimization function and the distance between every new observation and the origin. The computational complexity of our techniques mainly depends on solving a linear optimization problem, which is represented as $O(p)$, as well as computing covariance matrix, which is represented as $O(md^2)$. Hence, the maximum computational complexity of each node in EOOD and EAOD is $O(md^2)$, where m is the number of new observations to be classified. EBOD still needs to compute kernel matrix and the transformation of centered kernel function (especially for RBF kernel function), whose complexity is represented by $O(k)$. Thus the maximum computational complexity of EBOD for each node is $O(kmd^2)$.

The memory complexity of our techniques is mainly about keeping observations of the size of sliding window in memory being represented as $O(md)$, where d is the dimension of observations and m is the number of new observations to be classified. Overhead of storing other parameters such as covariance matrix with a complexity of $O(d^2)$ is negligible due to assuming $m > d$. Hence the maximum memory complexity of each node for our techniques is $O(md)$. Due to the fact that EBOD needs to keep $m \times m$ kernel function, its memory complexity of each node is $O(md + m^2)$. Table 6.3 summarizes these complexity.

6.7 Chapter Summary

In this chapter we have proposed distributed and online outlier detection techniques based on hyper-ellipsoid one-class SVM. We take into account data attribution correlation to precisely detect multivariate outliers. To cope with the problem of generating high false alarm rate, we also propose an updating strategy

Techniques	Communication Complexity	Computational Complexity	Memory Complexity
EBOD	–	$O(kmd^2)$	$O(md + m^2)$
SBOD	–	$O(kmp)$	$O(md + m^2)$
EOOD	$O(d^2)$	$O(md^2)$	$O(md)$
EAOD	$O(d^2)$	$O(md^2)$	$O(md)$
SAOD	$O(d)$	$O(mp)$	$O(md)$

Table 6.3: Complexity analysis of five outlier detection techniques for each sensor node

to incorporate new arrived observations and update the modelled hyper-ellipsoid SVM for more reliable outlier detection and detect the change of normal behavior of sensor data. We compare performance of our techniques with our online quarter-sphere SVM based techniques as well as two previously proposed batch techniques using both synthetic and real datasets as well as different labelling techniques. Experimental results show that our EAOD achieves better detection accuracy and lower false alarm. It implies that understanding data distribution and correlation among data attributes is essential to design a suitable outlier detection technique.

Chapter 7

Conclusions

In this chapter, we first provide a summary of the thesis. Then we present a list of our main research achievements and several important lessons learned from research on outlier detection in WSNs. We finally conclude our thesis by outlining future research directions in this area.

7.1 Thesis Overview

Chapter 1 introduces the motivation of outlier detection in WSNs and research objectives of this thesis. Motivated by the need to improve quality of data analysis and decision making, enhance efficiency of using WSNs resources by preventing unnecessary transmission of erroneous sensor observations, and increase effectiveness of monitoring and situation-awareness capabilities of the WSNs, this thesis focuses on online identification of outliers whenever and wherever they occur. We define outliers in WSNs as those observations that represent erroneous values (errors) or indicate particular phenomenal changes (events). Based on distributed in-network data processing, outlier detection techniques in this thesis identify sensor observations that do not conform to normal behavior of sensor data without using a pre-defined threshold or triggering conditions. We aim at designing and implementing effective and efficient outlier detection techniques for WSNs to identify outliers in an online and distributed manner and distinguish between errors and events with high accuracy and low false alarm, while maintaining the communication, computation and memory complexity low.

Chapter 2 provides a technique-based taxonomy of current outlier detection techniques developed for WSNs and provide a guideline on requirements of suit-

able outlier detection techniques for WSNs. First, several important issues that need to be considered while designing outlier detection techniques for WSNs, including sensor data characteristics, application-dependent issues, and performance matrices are presented. Based on these important considerations, general outlier detection techniques categorized based on techniques they use as well as their degree of using pre-labelled data are briefly reviewed and reasons why these general outlier detection techniques are not directly applicable to outlier detection in WSNs are explained. The taxonomy of current outlier detection techniques developed for WSNs includes statistical-based, nearest neighbor-based, clustering-based, classification-based and spectral decomposition-based techniques. Table 2.1 presents a comparison between state of the art on outlier detection techniques for WSNs to highlight their shortcomings. The guideline on main requirements of suitable outlier detection techniques for WSNs include (i) taking into account spatio-temporal and attribute correlations existing in sensor data, (ii) unsupervised and distributed identification of outliers in real-time, (iii) updating the modelled normal behavior of sensor data over time, (iv) making distinction between errors and events and appropriately handling them, and (v) having high detection accuracy while maintaining the resource consumption of WSNs to a minimum.

Chapter 3 investigates impact of data distribution and data dependencies on four data labelling techniques based on Mahalanobis distance, density, running average, and Baysian networks, and evaluates their performance for the outlier detection process. First, various types of outliers occurred in WSNs are illustrated, specifically incidental absolute errors, cluster absolute errors, random errors, and long-term errors and events. After detail explanation of principles of these four labelling techniques, a thorough comparison between these labelling techniques in terms of performance and complexity and the effect of the data characteristics on the labelling technique based on the real dataset collected at the Grand St. Bernard in Switzerland are presented. For the Grand St. Bernard dataset, in general, the Mahalanobis distance-based labelling technique is good for detecting outliers of Type 1, and outliers of Type 3 and 4 outside a solid dataset, when there are no extreme values. The density-based labelling technique is very useful for detecting outliers of Type 1 and 3. The running average-based labelling technique does not target a specific type of outliers and can detect all four types of outliers, but it will not find all the observations that intuitively can be identified as outliers.

Chapter 4 proposes statistical-based distributed and online outlier detection techniques based on using spatial and temporal correlations. Capturing spatial and temporal correlations of sensor data provides a better understanding about internal structure and characteristics of sensor data. Time series analysis is used

to model temporal correlation through the process of trend and seasonality analysis and removal as well as time series modelling and prediction. Geostatistical data analysis is used to model spatial correlation through the process of trend analysis, local spatial dependency modelling and geostatistical data prediction. We further alleviate high computational and memory complexity of modelling spatial and temporal correlations by ignoring seasonality, specifying AR model parameter, fitting the empirical variogram using a theoretical model, and making the modelling process local and distributed. Proposed temporal correlation-based, spatial correlation-based, and spatio-temporal correlations-based outlier detection techniques aim to enable each node to utilize predicted values estimated by the defined correlation models to classify their new sensor observations as either outlier or normal in an online manner and detect any changes in normal behavior of the sensor data upon occurrence. Experimental results on real environmental dataset collected at the Grand St. Bernard as well as different labelling techniques show high detection accuracy of our proposed outlier detection technique as well as their ability to detect changes in normal behavior of data.

Chapter 5 proposes distributed and online outlier detection techniques based on quarter-sphere one-class SVM and theory of spatial and temporal correlations to precisely detect outliers. To reduce high computational and memory complexity of quarter-sphere one-class SVM, we model the quarter-sphere SVM in the input space instead of the feature space. By using effective data preprocessing, log-transformation and autoscaling, the center of quartersphere is fixed at the origin in the input space. The basic decision function to determine whether a new arriving sensor observation is an outlier is further presented. To cope with the problem of generating high false alarm rate, three updating strategies are proposed to incorporate new arrived observations using a sliding window and updating the modelled quarter-sphere SVM for more reliable outlier detection and detecting the change of normal behavior of sensor data. The update strategies of the three techniques include updating the model (i) at each time interval, (ii) after a fixed-size time window, and (iii) depending on the previous decision results. We compare performance of our proposed quarter-sphere SVM based online and distributed techniques with a previously proposed batch technique using both synthetic and real datasets as well as different labelling techniques. Experimental results show that our technique SAOD has the ability to accurately detect outliers and the change of normal behavior in sensor data streams. It is also robust in terms of parameter selection, while keeping the communication, computational complexity and memory costs low.

Chapter 6 proposes distributed and online outlier detection techniques based on hyper-ellipsoid one-class SVM to identify multivariate outliers. Hyper-ellipsoid SVM differs from hyper-sphere SVM since hyper-ellipsoid SVM uses Mahalanobis

distance to determine the similarity of data vectors while hyper-sphere SVM uses Euclidean distance. Compared to Euclidean distance, Mahalanobis distance, which takes into account correlation between data attributes, has a better understanding of multivariate data structure as Euclidean distance is blind to attribute correlation and assumes all data vectors have equal distance from the center of mass. We simplify modelling the hyper-ellipsoid SVM in the input space and fix the centered of hyperellipsoid at the origin after data preprocessing. The basic decision function to determine whether a new arriving sensor observation is an outlier is further presented. To cope with the problem of generating high false alarm rate, we apply our update strategy to incorporate new arrived observations and update the modelled hyper-ellipsoid SVM for more reliable outlier detection and detect the change of normal behavior of sensor data. The performance our hyper-ellipsoid SVM based techniques are compared using both synthetic and real datasets as well as different labelling techniques. Experimental results show that the proposed technique EAOD achieves better detection accuracy and lower false alarm. It implies that understanding data distribution and correlation among data attributes is essential to design a suitable outlier detection technique.

7.2 Research Achievements

Our main research achievements can be classified as achievements related to concept of outliers and outlier detection techniques for WSNs:

- Achievements related to concept of outliers:
  - Identification of several important issues to be considered while designing outlier detection techniques for WSNs and providing a technique-based taxonomy of current outlier detection techniques developed for WSNs.
  - Explaining why general outlier detection techniques are not directly applicable to outlier detection in WSNs and providing a guideline on requirements that a suitable outlier detection technique for WSNs should meet.
  - Definition and identification of various types of outliers occurred in WSNs.
- Achievements related to outliers detection techniques:
  - Specifying the criteria to choose a labelling technique for performance evaluation of outlier detection techniques by extensive experiments using different labelling techniques on a real dataset.

- Proposing statistical-based distributed and online outlier detection techniques for WSNs based on quantification of spatial and temporal correlations. These proposed techniques enable each node to identify outliers and distinct between errors and events in real-time.
- Proposing distributed and online outlier detection techniques for WSNs from the machine learning and data mining perspective based on simplified models of quarter-sphere one-class SVM as well as hyper-ellipsoid one-class SVM. Designing update strategies to update the SVM-based model for more reliable outlier detection and detect the change of normal behavior of sensor data.

Looking back at our research objectives mentioned in Chapter 1, one can see that we have addressed both effectiveness and efficiency requirements of the optimal outlier detection technique for WSNs. High detection rate and low false alarm of our outlier detection techniques being applied on both synthetic and real datasets labelled with three different labelling techniques fulfills the requirement on effectiveness. The low communication overhead and computational and memory complexity of our distributed and online techniques fulfills the requirement on efficiency. Moreover, unlike existing outlier detection techniques for WSNs, we have integrated a mechanism to distinguish between errors and events in the outlier detection technique itself.

7.3 Lessons Learned

Important lessons we learned during this research can be summarized as:

- There is no universal outlier detection technique applicable for all application domains or data types. The design of an outlier detection technique is based on specific application requirements and corresponding data characteristics. Therefore, a suitable outlier detection technique should be specifically designed to meet sensor data characteristics and WSN requirements. This also implies that general outlier detection techniques from different research fields such as statistics, data mining and machine learning are not directly applicable for WSNs.
- Outliers have different types according to different classification criteria. For instance local outliers differ from global outliers in terms of size of dataset and number of sensor nodes used for outlier detection; spatial outliers differ from temporal outliers in terms on data dimension (space or time) taken into account; univariate outliers differ from multivariate outliers in terms

of number of data attributes used for outlier detection. Therefore, it is important that application requirements and outlier detection characteristics match.

- Two types of correlations may exist in sensor data, i.e, (i) correlation between data attributes, and (ii) spatio-temporal correlations between sensor data in WSNs. Capturing these correlations can provide a better understanding of data structure and improve the accuracy of outlier detection techniques. On the other hand, quantifying these correlations accurately may increase the resource consumption of WSNs due to high communication, computational and memory complexity. This implies that the trade-off between effectiveness and efficiency of outlier detection techniques should be carefully taken into account.
- Evaluation of outlier detection techniques is challenging due to the fact that for sensor data often no pre-labelled data is available. There is also no general purpose labelling technique. Different labelling techniques greatly impact performance evaluation of outlier detection techniques. Therefore, the characteristics of dataset as well as the labelling technique are the two deciding factors in selecting suitable labelling techniques for a certain dataset and outlier detection process.

7.4 Future Research Directions

In this thesis, we have proposed several distributed and online outlier detection techniques based on sensor data characteristics and WSN requirements. There are several important research directions related to outlier detection in WSNs, which need extra investigation. The list includes:

- Investigating alternative outlier detection techniques from statistical, data mining and machine learning communities.
- Alternative strategies to distinguish between errors and events, e.g., distance-based, or reputation-based techniques.
- Outlier detection for mobile WSNs, where the location of sensor nodes are changed over time.
- Outlier detection for heterogeneous data distributions, i.e., observations collected at a sensor node are drawn from multiple underlying distributions.
- Real implementation of outlier detection techniques on wireless sensor nodes.

Bibliography

[1] Ambient Systems. http://www.ambientsystems.net/ambient/index.html.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102-114, 2002.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Journal of Computer Networks*, 38(4):393-422, 2002.

[4] T. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 13rd Mediterranean Conference on Control and Automation*, pages 719-724, 2005.

[5] D. Barbara, C. Domeniconi, and J.P. Rogers. Detecting outliers using transduction and statistical significance testing. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 55-64, 2006.

[6] J. Branch, B. Szymanski, C. Giannella, and R. Wolff. In-network outlier detection in wireless sensor networks. In *Proceedings of IEEE ICDCS Conference*, 2006.

[7] L.A. Bettencourt, A. Hagberg, and L. Larkey. Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks. In *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems*, 2007.

[8] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD Conference on Managment of Data*, pages 93-104, 2000.

[9] S. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ACM SIGMOD Conference on Knowledge Discovery and Data*, pages 29-38, 2003.

[10] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons, New York, 1994.

[11] V. Bhuse and A. Gupta. Anomaly intrusion detection in wireless sensor networks. *Journal of High Speed Networks*, 15(1):33-51, 2006.

[12] A. Bronstein, J. Das, M. Duro, R. Friedrich, G. Kleyner, M. Mueller, S. Singhal, and I. Cohen. Bayesian networks for detecting anomalies in internet-based services. In *Proceedings of International Symposium on Integrated Network Management*, 2001.

[13] S. Brandes, I. Cosovic, and M. Schnell. *Introduction to Time-Series and Forecasting*. John Wiley and Sons, 1991.

[14] S. Basu and M. Meckesheimer. Antomatic outlier detection for time series: An application to sensor data. *Jounral of Knowledge and Information System*, 11(2):137-154, 2007.

[15] P.J. Brockwell and R.A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2002.

[16] A.L. Chiu and A.W. Fu. Enhancements on local outlier detection. In *Proceedings of International Database Engineering and Applications Symposium*, pages 298-307, 2003.

[17] J. Chen, S. Kher, and A. Somani. Distributed fault detection of wireless sensor networks. In *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, pages 65-72, 2006.

[18] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 2009.

[19] V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou, and B. Maglariset. Hierarchical anomaly detection in distributed large-scale sensor networks. In *Proceedings of ISCC Conference*, 2006.

[20] T. Clouqueur, K.K. Saluja, and P. Ramanathan. Fault tolerance in collaborative sensor networks for target detection. *IEEE Transactions on Computers*, 53(3), 2004.

[21] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and Hall/CRC, 2004.

[22] N.A.C. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, New York, 1991.

[23] C. Campbell and K.P. Bennett. A linear programming approach to novelty detection. *Advances in Neural Information Processing Systems*, 13:395-401, 2001.

[24] M. Conner. New battery technologies hold promise, peril for portable-system designers. *EDN Magazine*, 2005.

[25] M. Ding, D. Chen, K. Xing, and X. Cheng. Localized fault-tolerant event boundary detection in sensor networks. In *Proceedings of IEEE Conference of Computer and Communications Societies*, pages 902-913, 2005.

[26] M. Davy, F. Desobry, A. Gretton, and C. Doncarli. An online support vector machine for abnomal events detection. *Journal of Signal Processing*, 8(2):52-57, 2006.

[27] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[28] E. Elnahrawy and B. Nath. Context-aware sensors. In *Proceedings of EWSN Conference*, 2004.

[29] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of International Conference on Machine Learning*, pages 222-262, 2000.

[30] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of Knowledge Discovery and Data Mining*, pages 226-231, 1996.

[31] F.Y. Edgeworth. On discordant observations. *Philosophical Magazine*, 23(5):364-375, 1887.

[32] A. Foss and O. Zaane. A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets. In *Proceedings of International Conference on Data Mining*, pages 179-186, 2002.

[33] H. Fan, O.R. Zaiane, A. Foss, and J. Wu. A nonparametric outlier detection for effectively discovering top-n outliers from engineering data. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 557-566, 2006.

[34] J. Fu and X. Yu. Rotorcraft acoustic noise estimation and outlier detection. In *Proceedings of International Joint Conference on Neural Networks*, pages 4401-4405, 2006.

[35] Grubbs and Frank. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1-21, 1969.

[36] M.M. Gaber. Data stream processing in sensor networks. In *Learning from Data Streams Processing Techniques in Sensor Network (J. Gama and M. M. Gaber ed.)*, Springer Berlin Heidelberg, 2007.

[37] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 73-84, 1998.

[38] C.F. Garcia-Hernandez, P.H. Ibarguengoytia-Gonzalez, J. Garcia-Hernandez, and J.A. Perez-Diaz. Wireless sensor networks and applications: A survey. *International Journal of Computer Science and Network Security*, 7(3):264-273, 2007.

[39] J. Gehrke and S. Madden. Query processing in sensor networks. *IEEE Pervasive Computing*, 3(1):46-55, 2004.

[40] G.H. Golub, C.F.V, Loan. *Matrix Computations*. John Hopkins, 1996.

[41] GBR project. http://www.issnip.unimelb.edu.au.

[42] A. Gretton and F. Desobry. On-line one-class support vector machines: An application to signal segmentation. In *Proceedings of IEEE ICASSP*, 2003.

[43] D.M. Hawkins. *Identification of Outliers*. London: Chapman and Hall, 1980.

[44] D.J. Hill, B.S. Minsker, and E. Amir. Real-time bayesian anomaly detection for environmental sensor data. In *Proceedings of the 32nd Congress of the International Association of Hydraulic Engineering and Research*, 2007.

[45] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, 2006.

[46] T. Hu and S.Y. Sung. Detecting pattern-based outliers. *Official Publication of Pattern Recognition Letters*, 24(16):3059-3068, 2003.

[47] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85-126, 2003.

[48] Y. Hida, P. Huang, and R. Nishtala. Aggregation query under uncertainty in sensor networks. 2003.

[49] Z. He, X. Xu, and S. Deng. Discovering cluster based local outliers. *Official Publication of Pattern Recognition Letters*, 24(9-10):1651-1660, 2003.

[50] S. Harkins, H. He, G.J. Willams, and R.A. Baster. Outlier detection using replicator neural networks. In *Proceedings of International Conference on Data Warehousing and Knowledge Discovery*, pages 170-180, 2002.

[51] Intel Berkeley Research Laboratory. http://db.csail.mit.edu/labdata/labdata.html.

[52] ITU Internet Reports: The Internet of Things 2005, 7th edition. http://www.itu.int/publ/S-POL-IR.IT-2005.

[53] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle. Vision and Challenges for Realising the Internet of Things. Cluster of European Research Projects on the Internet of Things, 2010.

[54] D. Janakiram, A. Mallikarjuna, V. Reddy, and P. Kumar. Outlier detection in wireless sensor networks using Bayesian belief networks. In *Proceedings of IEEE Comsware Conference*, 2006.

[55] M.C. Jun, H. Jeong, and C.C.J. Kuo. Distributed spatio-temporal outlier detection in sensor networks. In *Proceedings of SPIE Conference*, 2006.

[56] M.F. Jiang, S.S Tseng, and C.M. Su. Tw-phase clustering process for outliers detection. *Official Publication of Pattern Recognition Letters*, 22(6-7):691-700, 2001.

[57] S.R. Jeffery, G. Alonso, M.J. Franklin, W. Hong, and J. Widom. Declarative support for sensor data cleaning. In *Processings of International Conference on Pervasive Computing*, pages 83-100, 2006.

[58] T. Johnson, I. Kwok, R.T. Ng. Fast computation of 2-dimensional depth contours. In *Proceedings of the ACM SIGMOD Conference on Knowledge Discovery and Data*, pages 224-228, 1998.

[59] W. Jin, A.K.H. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proceedings of the ACM SIGMOD Conference on Knowledge Discovery and Data*, pages 293-298, 2001.

[60] B. Krishnamachari and S. Iyengar, Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Transactions on Computers*, 53(3):241-250, 2004.

[61] E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large data sets. *International Journal of Very Large Data Bases*, pages 392-403, 1998.

[62] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold. Efficient biased sampling for approximate clustering and outlier detection in large data sets. *International Journal of Knowledge and Data Engineering*, 15(5):1170-1187, 2003.

[63] S. Kim and S. Cho. Prototype based outlier detection. In *Proceedings of International Joint Conference on Neural Networks*, pages 820-826, 2006.

[64] P. Kelly. An algorithm for merging hyperellipsodial clusters. *Technical report*, Los Alamos National Laboratory, 1994.

[65] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar. A comparative study of anomaly detection schemes in network intrusion detection. In *Processings of the SIAM Conference on Data Mining*, 2003.

[66] J. Laurikkala, M. Juhola, and E. Kentala. Informal identification of outliers in medical data. In *Proceedings of International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, 2000.

[67] X. Luo, M. Dong, and Y. Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55(1):58-70, 2006.

[68] P. Laskov, C. Schafer, and I. Kotenko. Intrusion detection in unlabeled data with quarter sphere support vector machines, *Detection of Intrusions and Malware & Vulnerability Assessment*, pages 71-82, 2004.

[69] C.E. Loo, M.Y. Ng, C. Leckie, and M. Palaniswami. Intrusion detection for routing attacks in sensor networks. *International Journal of Distributed Sensor Networks*, 2005.

[70] F. Martincic and L. Schwiebert. Distributed event detection in sensor networks. In *Proceedings of the International Conference on Systems and Networks Communication*, pages 43-48, 2006.

[71] M. Markos and S. Singh. Novelty detection: A review-part 1: Statistical approaches. *Journal of Signal Processing*, 83:2481-2497, 2003.

[72] M. Markos and S. Singh. Novelty detection: A review-part 2: Neural network based approaches. *Journal of Signal Processing*, 83:2499-2521, 2003.

[73] X. Ma, D. Yang, S. Tang, Q. Luo, D. Zhang, and S. Li. Online mining in sensor networks. In *Processings of IFIP International Conference on Network and Parallel Computing*, pages 544-550, 2004.

[74] Matlab. http://www.mathworks.com/products/matlab.

[75] S. Muthukrishnan, R. Shah, and J.S. Vitter. Mining deviants in time series data streams. In *Proceedings of International Conference on Scientific and Statistical Database Management*, 2004.

[76] Y. Meng and M.H. Dunham. Mining developing trends of dynamic spatiotemporal data streams. *Journal of Computers*, 1(3):43-50, 2006.

[77] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 491-502, 2003.

[78] M. Marin-Perianu. *Collaborative Wireless Sensor Networks in Industrial and Business Processes*. PhD thesis, University of Twente, 2008.

[79] K.V. Mardia and C.R. Goodall. Spatial-temporal analysis of mulitvariate environmental monitoring data. *Multivariate Environmental Statistics*, pages 347-386, 1993.

[80] S.G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGrawHill, 1996.

[81] S. Nittel, A. Labrinidis, and A. Stefanidis. *GeoSensor Networks*. Springer, 2006.

[82] K. Ni and G. Pottie. Sensor network data fault detection using hierarchical bayesian space-time modeling. *Technical report*, University of California, 2009.

[83] Meteosuisse. Federal Office of Meteorology and Climatology Switzerland. http://www.meteosuisse.admin.ch.

[84] E. Ould-Ahmed-Vall, G.F. Riley, and B.S. Heck. Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks. *Georgia Institute of Technology*, 2006.

[85] I. Onat and A. Miri. An intrusion detection system for wireless sensor networks. In *Proceeding of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 253-259, 2005.

[86] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *International Joural of CACM*, 47(6):53-57, 2004.

[87] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Processings of International Conference on Data Engineering*, pages 315-326, 2003.

[88] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Distributed deviation detection in sensor networks. In *ACM Special Interest Group on Management of Data*, pages 77-82, 2003.

[89] Y. Panatier. *Variowin: Software for Spatial Data Analysis in 2D*. Springer Berlin Heidelberg, New York, 1996.

[90] M.I. Petroveskiy. Outlier detection algorithms in data mining system. *Journal of Programming and Computer Software*, 29(4):228-237, 2003.

[91] A.V.U. Phani, A. Mallikarjuna, and D. Janakiram. Distributed collaboration for event detection in wireless sensor networks. In *Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing*, pages 1-8, 2005.

[92] G.J. Pottie and W.J. Kaiser. Wireless integrated network sensors. *Communication of ACM*, 43(5):51-58, 2000.

[93] D. Pokrajac, A. Lazarevic, and L.J. Latecki. Incremental local outlier detection for data streams. In *Processings of IEEE Symposium on Computational Intelligence and Data Mining*, 2007.

[94] D. Ren, I. Rahal, and W. Perrizo. A vertical outlier detection algorithm with clusters as by-product. In *Proceedingd of International Conference on Tools with Artificial Intelligence*, pages 22-29, 2004.

[95] I. Ruts and P. Rousseeuw. Computing depth contours of bivariate point clouds. *Journal of Computational Statistics and Data Analysis*, 23:153-168, 1996.

[96] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, 1996.

[97] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Processings of ACM Special Interest Group on Management of Data*, pages 427-438, 2000.

[98] S. Rajasegarar, C. Leckie, M. Palaniswami, and J.C. Bezdek. Distributed anomaly detection in wireless sensor networks. In *Proceedings of IEEE ICCS Conference*, 2006.

[99] S. Rajasegarar, C. Leckie, M. Palaniswami, and J.C. Bezdek. Quarter sphere based distributed anomaly detection in wireless sensor networks. In *Proceedings of IEEE International Conference on Communications*, pages 3864-3869, 2007.

[100] S. Rajasegarar, C. Leckie, and M. Palaniswami. CESVM: Centered hyperellipsoidal support vector machine based anomaly detection. In *Proceedings of IEEE International Conference on Communications*, pages 1610-1614, 2008.

[101] R project. http://www.r-project.org.

[102] B. Sheng, Q. Li, W. Mao, and W. Jin. Outlier detection in sensor networks. In *Proceedings of MobiHoc Conference*, 2007.

[103] P. Sun. *Outlier Detection in High Dimensional, Spatial and Sequential Data Sets*. PhD thesis, University of Sydney, Sydney, 2006.

[104] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogerakiand, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. *Journal of Very Large Data Bases*, 2006.

[105] P. Sykacek. Equivalent error bars for neural network classifiers trained by bayesian inference. In *Proceedings of European Symposium on Artificial Neural Networks*, pages 121-126, 1997.

[106] B. Scholkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high dimensional distribution. *Journal of Neural Computation*, 13(7):1443-1471, 2001.

[107] A.A. Sebyala, T. Olukemi, and L. Sacks. Active platform security through intrusion detection using naive bayesian network for anomaly detection. In *Proceedings of Communications Symposium*, 2002.

[108] SensorScope Sytem. http://sensorscope.epfl.ch/index.php/Main_Page.

[109] K. Shih, S. Wang, P. Yang, and C. Chang. CollECT: Collaborative event detection and tracking in wireless heterogeneous sensor networks. In *Proceedings of the 11st IEEE Symposium on Compuers and Communications*, pages 935-940, 2006.

[110] A.P.R. Silva, M.H.T. Martins, B.P.S. Rocha, A.A.F. Loureiro, L.B. Ruiz, and H.C. Wong. Decentralized intrusion detection in wireless sensor networks. In *Proceedings of the first ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks*, pages 16-23, 2005.

[111] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Andestrin. Habitat monitoring with sensor networks. *Communnication of ACM*. 47:34-40, 2004.

[112] S.K. Sahu, G.J. Lasinio, A. Orasi, and K.V. Mardia. A comparison of spatio-temporal bayesian models for reconstruction of rainfall fields in a cloud seeding experiment. *Journal of Mathematics and Statistics*, 1(4):273-281, 2005.

[113] I. Solis and K. Obraczka. Isolines: Efficient spatio-temporal data aggregation in sensor networks. *Wireless Communications and Mobile Computing*, 9(3):357-367, 2009.

[114] G. Sterk and A. Stein. Mapping wind-blown mass transport by modeling variability in space and time. *Soil Science Society of America Journal*, pages 232-239, 1997.

[115] R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications: with R Examples*. Springer Berlin Heidelberg, 2006.

[116] M.L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York, 1999.

[117] J. Tukey. *Exploratory Data Analysis*. Addison Wesley, 1997.

[118] P.N. Tan, M. Steinback, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[119] P.N. Tan. Knowledge discovery from sensor data. *Sensors*, 2006.

[120] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mobile Computing and Communications Review*, 6(2):28-36, 2002.

[121] D.M.J. Tax and R.P.W. Duin. Support vector domain description. *Official Publication of Pattern Recognition Letters*, 20:1191-1199, 1999.

[122] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Journal of Machine Learning*, 54(1):45-56, 2004.

[123] M. Tubaishat and S. Madria. Sensor networks: An overview. *IEEE Potentials*, 22(2):20-23, 2003.

[124] P.M. Valero-Mora, F.W. Young, and M. Friendly. Visualizing categorical data in ViSta. *Journal of Computational Statistics & Data Analysis*, 43:495-508, 2003.

[125] M.C. Vuran, O.B. Akan, and I.F. Akyildiz. Spatio-temporal correlation: Theory and applications for wireless sensor networks, *Journal of Computer and Telecommunications Networking*, 45(3):245-259, 2004.

[126] M.C. Vuran and O.B. Akan. Spatio-temporal characteristics of point and field source in wireless sensor networks. In *Proceedings of IEEE ICC*, pages 234-239, 2006.

[127] C.T. Vu, R.A. Beyah, and Y. Li. Composite event detection in wireless sensor networks. In *Proceedings of Performance, Computing, and Communications Conference*, pages 264-271, 2007.

[128] V.N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[129] K. Varmuza and P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, 2009.

[130] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):66-75, 1991.

[131] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, and P. Deng. Localized outlying and boundary data detection in sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1145-1157, 2007.

[132] D. Wang, D.S. Yeung, E.C.C Tsang. Structured one-class classification. *IEEE Transactions on System, Man and Cybernetics*, 36(6):1283-1295, 2006.

[133] X.H. Wang, X.N. Zhou, P. Vounatsou, Z. Chen, J. Utzinger, K. Yang, P. Steinmann, and X.H. Wu. Bayesian spatio-temporal modeling of schistosoma japonicum prevalence data in the absence of a diagnostic gold standard. *Journal of Parasitology*, 2(6):1-9, 2008.

[134] C.K. Wikle, L.M. Berliner, and N. Cressle. Hierarchical Bayesian Space-Time Models. *Journal of Environmental and Ecological Statistics*, 2(5):117-154, 1998.

[135] R. Webster, M.A. Oliver. *Geostatistics for Environmental Scientists*. Springer Berlin Heidelberg, 2007.

[136] D. Yu, G. Sheikholeslami, and A. Zhang. Findout: Finding outliers in very large datasets. *Journal of Knowledge and Information Systems*, 4(3):387-412, 2002.

[137] Y. Yao and J. Gehrke. Query processing for sensor networks. In *Proceedings of the CIDR Conference*, 2003.

[138] K. Zhang, S. Shi, H. Gao, and J. Li. Unsupervised outlier detection in sensor networks using aggregation tree. In *Proceedings of ADMA Conference*, 2007.

[139] M. Zoumboulakis and G. Roussos. Escalation: Complex event detection in wireless sensor networks. *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pages 270-285, 2007.

[140] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 103-114, 1996.

[141] Y. Zhuang and L. Chen. In-network outlier cleaning for data collection in sensor networks. In *Proceedings of VLDB Conference*, 2006.

[142] M. Zoumboulakis and G. Roussos. Escalation: Complex event detection in wireless sensor networks. *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pages 270-285, 2007.

[143] D. Zeinalipour-Yazti, P.K. Chrysanthis. Mobile Sensor Network Data Management. In *Encyclopedia of Database Systems (M.T. Ozsu and L. Liu eds)*, 2009.

[144] Y. Zhang, N. Meratnia, and P.J.M. Havinga. A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets. *Technical Report*, University of Twente, 2007.

[145] Y. Zhang, N. Meratnia, and P.J.M. Havinga. Why general outlier detection techniques do not suffice for wireless sensor networks? *Intelligent Techniques for Warehousing and Mining Sensor Network Data (C. A ed.)*. IGI Global, 2009.

[146] Y. Zhang, N. Meratnia, and P.J.M. Havinga. Outlier detection techniques for wireless sensor network: A survey. *IEEE Communications Surveys & Tutorials*, 12(2):159-170, 2010.

[147] Y. Zhang, N. Meratnia, and P.J.M. Havinga. An online outlier detection technique for wireless sensor networks. In *Proceedings of the Third IEEE European Conference on Smart Sensing and Context (EuroSSC)*, pages 25-26, 2008.

[148] Y. Zhang, N. Meratnia, and P.J.M. Havinga. An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine. In *Proceedings of the Fourth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 151-156, 2008.

[149] Y. Zhang, N. Meratnia, and P.J.M. Havinga. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In *Proceedings of the IEEE 23rd International Conference on Advanced Information Networking and Applications Workshops/Symposia*, pages 990-995, 2009.

[150] Y. Zhang, N. Meratnia, and P.J.M. Havinga. Ensuring high sensor data quality through use of online outlier detection techniques. *Special Issue on Data Quality Management in Wireless Sensor Networks of International Journal of Sensor Networks (IJSNet)*, 7(3):141-151, 2010.

Publications

Journals

- Y. Zhang, N. Meratnia, and P.J.M. Havinga. Outlier detection techniques for wireless sensor network: A survey. *IEEE Communications Surveys & Tutorials*, 12(2):159-170, 2010.
- Y. Zhang, N. Meratnia, and P.J.M. Havinga. Ensuring high sensor data quality through use of online outlier detection techniques. *Special Issue on Data Quality Management in Wireless Sensor Networks of International Journal of Sensor Networks (IJSNet)*, 7(3):141-151, 2010.

Book chapter

- Y. Zhang, N. Meratnia, and P.J.M. Havinga. Why general outlier detection techniques do not suffice for wireless sensor networks? *Intelligent Techniques for Warehousing and Mining Sensor Network Data (C. A ed.)*, pages 136-158, IGI Global, 2009.

Peer reviewed conferences & workshops

- Y. Zhang, N. Meratnia, and P.J.M. Havinga. Hyperellipsoidal support vector machine-based online outlier detection technique for geosensor networks. In *proceedings of The Third International Conference on Geosensor Networks (GSN)*, pages 31-41, 2009.
- Y. Zhang, N. Meratnia, and P.J.M. Havinga. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In *Proceedings of the IEEE 23rd International Conference on*

Advanced Information Networking and Applications Workshops/Symposia, pages 990-995, 2009.

- M. Bahrepour, Y. Zhang, N. Meratnia, and P.J.M. Havinga. Use of event detection approaches for outlier detection in wireless sensor networks. In *Proceedings of the Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 439-444, 2009.

- Y. Zhang, N. Meratnia, and P.J.M. Havinga. An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine. In *Proceedings of the Fourth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 151-156, 2008.

- Y. Zhang, N. Meratnia, and P.J.M. Havinga. An online outlier detection technique for wireless sensor networks. In *Proceedings of the Third IEEE European Conference on Smart Sensing and Context (EuroSSC)*, 2008.

- Y. Zhang, S. Chatterjea, and P.J.M. Havinga. Experiences with implementing a distributed and self-organizing scheduling algorithm for energy-efficient data gathering on a real-life sensor network platform. In *Proceedings the First IEEE International Workshop on From Theory to Practice in Wireless Sensor Networks*, IEEE Computer Society, 2007.

- S. Chatterjea, T. Nieberg, Y. Zhang, and P.J.M., Havinga. Energy-efficient data acquisition using a distributed and self-organizing scheduling algorithm for wireless sensor networks. In *Proceedings of the Third IEEE Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 368-385, Lecture Notes in Computer Science, Springer Verlag, 2007.

- Y. Zhang, J. Wu, and P.J.M. Havinga. Implementation of an on-demand routing protocol for wireless sensor networks. In *Processings of 13th International Conference on Telecommunications (ICT)*, 2006.