11-12-2013

Novel Online Data Cleaning Protocols for Data Streams in Trajectory, Wireless Sensor Networks

Sitthapon Pumpichet
spump001@fiu.edu

DOI: 10.25148/etd.FI13120904

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

NOVEL ONLINE DATA CLEANING PROTOCOLS FOR DATA STREAMS IN TRAJECTORY, WIRELESS SENSOR NETWORKS

A dissertation submitted in partial fulfillment of

the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Sitthapon Pumpichet

2013

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Sitthapon Pumpichet, and entitled Novel Online Data Cleaning Protocols for Data Streams in Trajectory, Wireless Sensor Networks, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Stavros Georgakopoulos

Deng Pan

Nikolaos Tsoukias

Syed M. Ahmed

Niki Pissinou, Major Professor

Date of Defense: November 12, 2013

The dissertation of Sitthapon Pumpichet is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2013

DEDICATION

To my family,

Pisuth, Panthipa and Pornnapa Pumpichet,

and all of my merciful teachers.

ACKNOWLEDGMENTS

This dissertation not only represents my effort in research, but also a significant milestone in my life. I, myself alone, cannot accomplish this dissertation. Every step to this success has been made by supportive environments and people at Florida International University.

I am grateful to my major advisor, Professor Niki Pissinou, the most important person in my research profession. She is one of the smartest sophisticates I have ever worked with. She is someone you will feel amazed on how insightful she is able to perceive her students. She has mentored me with care and mercy for five years. She has encouraged me when I am down, guided me when I am lost, and support me in all aspects from the beginning until the final stage of this dissertation. Her advice and mentorship have guided me to succeed in conducting independent and ethical research. I am truly honored to have an opportunity to work with Professor Niki Pissinou.

I would like to express my deep gratitude to Dr. Stavros Georgakopoulos, Dr. Deng Pan, Dr. Nikolaos Tsoukias and Dr. Syed Ahmed, my dissertation committee members, for serving on my dissertation and always giving me enlightening comments. I also would like to show my appreciation to Dr. Kia Makki, Dr. Khokiat Kengskool and Dr. Jean Andrian for merciful guiding me throughout my doctoral study.

Thanks to Florida International University, Doctoral Evidence Acquisition Fellowship, Graduate and Professional Student Committee Scholarship, as well as Graduate, Teaching and Research Assistantships allow me to have opportunity to have all kinds of resources for this accomplishment. Especially thanks to Dr. Shekhar Bhansali, Dr. Kang Yen, Ms. Maria Benincasa and all administrators in the Department of

Electrical and Computer Engineering, the School of Computer and Information Sciences, and The Dean's office of College of Engineering and Computing for supporting me in all aspects until the last stage of my doctoral study.

Besides, I would like to thank to the Telecommunication and Information Technology Institute (IT2) where I encountered a collaborative and enthusiastic research environments. Thanks to all colleagues in IT2 and the Department of Electrical and Computer Engineering. Special thanks to Dr. Xinyu Jin, Dr. Qutub Bakhtiar, Dr. Qian Wang, Dr. Charles A Kamhoua, Dr. Hao Jin, Dr. Shan Jiang and Dr. Kai Chen.

Finally, beyond words I can express my gratitude to my family, Panthipa, Pisuth and Pornnapa Pumpichet. Their unconditional love, support and belief always make me overwhelmed and have carried me through all the hard times. This accomplishment is theirs.

ABSTRACT OF THE DISSERTATION

NOVEL ONLINE DATA CLEANING PROTOCOLS FOR DATA STREAMS IN TRAJECTORY, WIRELESS SENSOR NETWORKS

by

Sitthapon Pumpichet

Florida International University, 2013

Miami, Florida

Professor Niki Pissinou, Major Professor

The promise of Wireless Sensor Networks (WSNs) is the autonomous collaboration of a collection of sensors to accomplish some specific goals which a single sensor cannot offer. Basically, sensor networking serves a range of applications by providing the raw data as fundamentals for further analyses and actions. The imprecision of the collected data could tremendously mislead the decision-making process of sensor-based applications, resulting in an ineffectiveness or failure of the application objectives. Due to inherent WSN characteristics normally spoiling the raw sensor readings, many research efforts attempt to improve the accuracy of the corrupted or “dirty” sensor data. The dirty data need to be cleaned or corrected. However, the developed data cleaning solutions restrict themselves to the scope of static WSNs where deployed sensors would rarely move during the operation.

Nowadays, many emerging applications relying on WSNs need the sensor mobility to enhance the application efficiency and usage flexibility. The location of deployed sensors needs to be dynamic. Also, each sensor would independently function and contribute its resources. Sensors equipped with vehicles for monitoring the traffic condition could be depicted as one of the prospective examples. The sensor mobility causes a transient in network topology and correlation among sensor streams. Based on static relationships among sensors, the existing methods for cleaning sensor data in static WSNs are invalid in such mobile scenarios. Therefore, a solution of data cleaning that considers the sensor movements is actively needed.

This dissertation aims to improve the quality of sensor data by considering the consequences of various trajectory relationships of autonomous mobile sensors in the system. First of all, we address the dynamic network topology due to sensor mobility. The concept of virtual sensor is presented and used for spatio-temporal selection of neighboring sensors to help in cleaning sensor data streams. This method is one of the first methods to clean data in mobile sensor environments. We also study the mobility pattern of moving sensors relative to boundaries of sub-areas of interest. We developed a belief-based analysis to determine the reliable sets of neighboring sensors to improve the cleaning performance, especially when node density is relatively low. Finally, we design a novel sketch-based technique to clean data from internal sensors where spatio-temporal relationships among sensors cannot lead to the data correlations among sensor streams.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

Sensor devices of ever-increasing versatility and decreasing size are encasing our world. They are crowding into embedded systems, tele-care designs, and are deployed in increasing ubiquity in sensor networks of every type. While the advent of sensors has increased a societal dependence on a continued availability and reliability of data, this dependence has been significantly enhanced by the new wireless paradigms where plenty of mobile sensors communicate with each other to form mobile Wireless Sensor Networks (mWSNs).

Sensors in mWSNs typically generate high volumes of data streams that are anticipated to be used by applications that require a real-time response. It is clear, however, that sensors which inherit limitations of low-power wireless transceiver units, limited memory and computational capacity do not gather or forward accurate data at all times. Interference and congestion alone minimize the quality of the data collected at a base station or a point of data acquisition. Degraded quality of the received sensor data lowers the service performance of mWSN applications. Therefore, mechanisms to clean sensor data which improve quality of sensor data are mandatory.

Since sensor nodes normally deployed in mWSNs operate in an unattended fashion and have disposable and irreplaceable power sources, any mechanisms including data cleaning mechanisms that require processes in sensor nodes need to be well designed in prolonging the operational sensor lifetime. Meanwhile, most existing data cleaning approaches are designed for applications in which sensors' locations are unchanged. They do not consider the overall cost of data cleaning in an environment that possibly includes hundreds or thousands of moving sensors. Instead, an effective data cleaning in such a mobile setting needs to recognize the sensor mobility while preserving power consumption of the network. In this research, we studied and developed efficient techniques to clean sensor data in a centralized scheme to reduce the processing load and power consumption spent in a sensor node.

1.1 Motivating Applications

Promising applications relying on mWSNs range in various fields, such as security, transportation and healthcare systems. These applications become more seamless and bring benefits to our daily lives. They need to correctly perform analyses based on the collected sensor data and response in a real-time fashion, although the sensors are remote from the data acquisition center. The imprecision of collected sensor data could mislead analyzers to incorrect action plans and cause unexpected losses. For example, in the vehicular system domain, a traffic center can analyze data streams from sensors attached to vehicles. This capability could support real time analysis of traffic conditions around a specific area, which in turn could activate a notification alert to the other vehicles for a better route. Due to data imprecision collected from sensors, the traffic analyzer could misinterpret the traffic condition and mislead other vehicles to detour into a wrong path and cause them an unnecessary waste of gas consumption or even bring them an unexpected accident risk.

Another scenario is patient monitoring. A sensor based insulin pump could be used to continuously monitor the sugar level of a diabetic patient, auto-detect the glucose content in the blood and even inject the exact dosage of insulin into the patient. Sensors can be placed in an artery to measure the blood pressure of an elderly patient. Moreover, sensors embedded in the skull of an Alzheimer patient can be used to monitor and regulate the patient's neuron activity and other vital signs. Similarly, a nanotube biomedical sensor can be attached to a patient with asthma symptoms to monitor electronic variations caused by the presence of asthma nitric oxide and other airborne asthma pathogens when patients move from place to place. The sensor alerts the patient immediately when she/he moves into areas where the level of pathogens exceeds the specified threshold. Simultaneously, the sensor sends the stream of pathogen levels and trajectory data to the health station, where the data will be further processed in order to alert other patients in real-time for highly precipitated areas. Without data cleaning mechanisms, the poor quality of sensor data received at the healthcare center could not only lead to incorrect treatments and extra medical costs, but it can even harm patients to death.

1.2 Research Needs and Challenges

The motivating examples mentioned above not only demonstrate the significance of this study, but also pinpoint a major challenge that needs to be tackled before a feasible solution is realized. In particular, a main and common problem of these applications is that the location of the sensor nodes keeps changing. It means that the existing data cleaning methods that rely on static group of associated sensor nodes would be unavailable for these scenarios. The practical solution of data cleaning in mWSNs needs to consider mobility of sensor nodes and dynamic topology of sensor networks. While researchers have attempted to improve the quality of the received sensor data, little work has been done in the field of mWSNs.

In works [AFM06,CFG$^+$07,MNP10], the goal is to correct or "clean" the corrupted data statically stored in the databases. These techniques require an access to a complete set of the data that are stored in the databases in order to formulate comprehensive data cleaning solutions. Dealing with a high volume of the stored data, these methods cannot provide a timely response to time-sensitive continuous queries inherent in mWSNs; therefore, they are not applicable to process data streams in sensor applications. Furthermore, most approaches [JAF$^+$06,PS07,EN04] to clean sensor streams assume that the sensor locations are static and the contextual relationships in time and space among sensors remain unchanged. Such techniques cannot be applied to mWSNs, where the sensors keep moving and creating dynamic contextual relationships among sensors.

Some works view mobile sensors as geometries (e.g., points, lines, areas, volumes) [LHW07,ABN08] changing over time. To record the position of a mobile sensor, these works define a sensor's physical movement as a "trajectory" that denotes the evolving position of the sensor from its initial position to its final position. They therefore define a trajectory as a function of space and time. In reality, however, the description, representation and manipulation of a mobile sensor's trajectory are more complicated. Furthermore, although some works [NYZ12,GTW$^+$10] reduce the volume of a mobile sensor's trajectory and sensor readings, they do not focus on cleaning noisy and corrupted sensor data. In the face of volumes of data streams transmitted by moving sensors displaying various degrees of precision, accuracy, and dynamism, the core challenge is to identify techniques that model and infer situations from mobile sensor data that can be used in a data cleaning process. Clearly, much work remains to be done to clean sensor data streams in mWSNs.

1.3 Research Objectives

In mWSN applications, sensor data are prone to data missing and all types of data corruption. The more we can clean or correct the dirty data, the more feasible the applications in mWSNs are. This research stems from the recognition that applications of mWSNs will remain elusive unless a solution that incorporates dynamism of sensor topology into a data cleaning mechanism is developed. This research involves the design, development and experimental demonstration of an online data stream cleaning methodology for mWSNs that incorporates dynamism of spatio-temporal relationships among mobile sensors. In particular, we studied and investigated the following topics.

1. Data cleaning in dense networks

Indeed, the sensor selection process is a primary challenge that is not well tackled in existing sensor data cleaning methods. In particular, existing methods rely on an associated set of static sensors for their cleaning processes. However, when sensors are moving, we cannot rely on a predefined static set of sensors. Therefore, these methods are not transferable to mWSNs. Our hypothesis is that we can select the most helpful set of neighboring sensors to clean data of a sensor if we can characterize the dynamic relationship of sensors in a specific area. The higher the density of sensor nodes in the area of interest, the easier a set of helpful sensors can be selected. Therefore, our first objective is to develop a data cleaning method in dense networks by characterizing the dynamic relationships of mobile sensors in a pre-defined area.

2. Data cleaning in sparse networks

According to the developed method for dense networks, we found that it is more difficult to select a set of neighboring sensors that is helpful for cleaning data when the density of mobile sensors in an area of interest is lower. Our hypothesis is that mobile sensors which have highly relative trajectories are helpful in cleaning data for each other. In trajectory analysis, existing solutions represent a sensor trajectory as an ordered list of location samples at specific instances in time [GS05]. Such a list can express the changing positions of an object throughout its lifespan but it does not contain information necessary to extract the sensor trajectories into meaningful trajectory relationships. Therefore, the objective of this task is to develop a data cleaning in sparse networks by exploiting a relative trajectory pattern among mobile sensors.

3. Data cleaning for internal sensing

In many scenarios of mWSN applications, mobile sensors are not sensing a shared physical phenomenon in environments. These types of sensors include, but are not limited to, a measurement device for glucose level, heart rate and blood pressure of a patient, or gas consumption, speed and acceleration of a vehicle. The measurement from these types of sensors could change abruptly and not be likely to demonstrate a correlation to those of neighboring sensors. For such scenarios, we aim to develop a data cleaning scheme that does not rely on spatio-temporal or trajectory relationships among mobile sensors.

1.4 Research Approaches

We first investigate networks where there are a number of sensors are moving in the system. To clean data in a dense network, we developed the Virtual static Sensor (VS) concept to help clean data from mobile sensors. We considered that sensors are traversing across areas with diverse environments resulting in different levels of sensor readings. We assumed that boundaries of areas with different sensing levels are known and all forms of dirty data have been detected. This detected set of dirty data would be discarded and treated as a missing data before being forwarded to our designed data cleaning module.

The developed data cleaning method deploys a concept of Virtual static Sensor (VS) to reinstate the missing data in mWSNs. Before the cleaning process begins, each VS will be assigned with the location and the coverage area. Each of them will not move; it will observe the sensor data by processing the measurements physically sensed by other moving sensors in its coverage area. The VS knows best, compared to moving sensors, about the measurement trend in its nearby area. Therefore, the VS is helpful for the cleaning process to estimate missing data of real sensors in its vicinity. As an analogy, a VS acts as a host who knows best about events happening around her home and can provide, if needed, information around her home to her visitors. The developed VS concept cooperates with a traditional linear adaptive filter in prediction mode to increase the cleaning performance.

Since the performance of the VS method is significantly decreased when sensor nodes in the network are sparse, we also developed a cleaning method that performs well within a sparse network. This method also considers the sensor trajectory relative to a pre-defined area. It would select the most helpful and reliable moving sensors and deploy their data into the cleaning module. The selection mechanism is based on the sensing consistency of each sensor stream for a pre-defined area. Like our previous design, it is assumed that all forms of corrupted data are detected before being forwarded to our proposed data cleaning module. Again, the designed cleaning process is a centralized based architecture, i.e., all cleaning mechanisms including the detection of dirty data and data stream management are conducted at the base station, where all sensor data streams are forwarded. The trajectory data and the sensor readings from a sensor could be delivered to the base station via different channels as an out-of-bound transmission. Although the sensor measurements are dirty and need to be cleaned, we assume that the trajectory data is correctly received at the base station.

This developed data cleaning method is an area-based approach, assuming a priori knowledge of sub-area boundaries. The cleaning process computes the replacement of dirty data by utilizing the readings from a group of sensors that are believed to be offering enough consistent readings from a specific sub-area. Based on a priori knowledge of sub-area boundaries, each sub-area has been indexed and matched with a belief table. Our approach uses the belief table, which contains the updated belief degree of each sensor for each sub-area. For a sub-area, the belief degree of each sensor represents the consistency and reliability level of readings from a sensor that could help in cleaning dirty samples collected from the corresponding sub-area at a time.

However, there are many types of sensors that do not measure a surrounding phenomenon, for example, sensors measuring heart rate, blood pressure and glucose level of patients, or sensors gauging acceleration and gas level of vehicles, etc. The readings from these sensors are not likely correlated although the locations and trajectories of these sensors are close. We are motivated by these common scenarios, where the existing online data cleaning techniques including our developed methods are not applicable. In this case, we further developed a sketch-based data cleaning method that can clean data streams of sensors in such environments. The sketch-based method is a pairwise cleaning between the base station and a corresponding mobile sensor. Each sensor exploits the unique characteristic of a super-increasing set to formulate a sketch packet, which can help clean *N* samples of sensor data once received at the base station.

1.5 Research Contributions

According to the research objectives and approaches, we have developed comprehensive data cleaning solutions that include protocol design, algorithm development, experimental and simulation proofs and analyses. Specifically, we make the following contributions.

1. **Developed the virtual static sensor concept to clean sensor streams in dense networks.**

To clean sensor data in dense networks, we developed a novel data cleaning method using the concept of virtual static sensor. In this work, our main contributions are listed as follows.

- We developed a concept of virtual static sensor to collect data from mobile sensors which have close spatio-temporal relationships within a coverage area of a virtual sensor. To our best knowledge, this method is the first data cleaning method that is designed and validated to mWSNs, where network topology among sensors is transient.

- We applied the use of adaptive filter in prediction mode by combining it with virtual sensor to reinstate the value of missing sensor data in mWSNs.

2. **Developed the belief-based data cleaning method for sparse networks.**

To overcome a limitation of the virtual sensor based method, we focused on investigating how to clean sensor data streams in sparse networks. In this work, our main contributions are as follows.

- We introduced a belief-based sensor selection method to identify the group of sensors that is helpful in cleaning data based on their current trajectories and the quality of their data streams.
- We developed a novel online data cleaning method designed for the dynamic environment in mWSN applications. Our evaluation results show that the cleaning performance of this method outperforms those of the virtual sensor-based method and an existing method designed for stationary sensor networks.

3. **Developed the sketch-based data cleaning method for internal sensing environments.**

In scenarios where sensors are not measuring a shared environment, we developed a novel sketch-based data cleaning method. The main contributions of this work are listed as follows.

- We developed a sketch-based data cleaning method for data streams in mWSNs where moving sensors do not measure a shared phenomenon and when they are also deployed in a sparse network.
- We exploit a unique characteristic of a super-increasing set to help formulate a sketch packet which plays a key role in cleaning data when received at the base station.

1.6 Organization of the Dissertation

Up to this point, we have introduced the background, challenges, research objectives and approaches of this dissertation. The remainder of this dissertation is organized as follows. First, we review the comprehensive literature works related to data cleaning in mWSNs. We then present the use of the virtual sensor concept for cleaning sensor data and its justification in Chapter 3. We describe the belief-based cleaning method that analyzes patterns of sensor movements relative to sensing boundary in Chapter 4. In Chapter 5, we discuss the novel cleaning method based on a sketch technique. Finally, we conclude what we achieved and project the potential research directions in Chapter 6.

CHAPTER 2

RELATED WORKS

Data cleaning is a pre-process that is commonly used to reduce all types of data imprecision. Types of data imprecision include, but are not limited to, data missing, noisy data, data misplacement and data replication, etc. The suitable technique for data cleaning is application dependent. That is, there is no generic mechanism for cleaning data in all types of applications.

Data cleaning is firstly used in applications involving databases with forms of data imprecision. For more than a decade, researchers have focused primarily on cleaning data in the static databases or data warehouse. Applications of cleaning databases include outlier detection and replacement [SPP$^+$06], data consistency maintenance [CFG$^+$07] and reduction of the data uncertainty [CCX08]. To accomplish such tasks, existing approaches require access to a complete set of static databases. However, deployment of sensors in a domain normally generates a very high volume of streaming data, while most sensor applications need a timely response through types of continuous queries [TM06,Agg02]. Storing the whole received sensor data and then processing them in a static database cannot respond to this requirement of sensor applications. Therefore, current solutions to data cleaning are not suitable for typical WSN and mWSN applications.

The purpose of this chapter is to provide the survey on data cleaning mechanisms, frameworks, and architectures in both static and mobile Wireless Sensor Networks (WSNs) and to discuss how they are related to this research as follows.

2.1 Data Cleaning in Static Wireless Sensor Networks

2.1.1 Sequential based methods

The first sequential based data cleaning method for WSN was proposed in [JAF$^+$06]. This work was the collaboration of a research group that is responsible for developing the Stanford data stream management system (STREAM) [ABB$^+$03]. In [JAF$^+$06], authors proposed a data cleaning framework, Extensible Sensor stream Processing (ESP), which is applicable for both RFID and WSN data. This work was designed to be integrated with the STREAM data stream management system. ESP data cleaning framework, as shown in Fig 2.1, proposed the cleaning operation with a cascade of five programmable stages: Point, Smooth, Merge, Arbitrate and Virtualize. All five of these stages are not necessary to clean sensor data for a given application. Basically, the objective of the Point stage is to screen the individual readings that conflict with the predicated system rules, for example, clear distanced based outliers. The Smooth stage aims to clean the missed readings in a single data stream based on temporal relationships of the data sequence. In the Merge stage, the cleaning process needs to use spatial properties of at least one stream to correct the data. The Arbitrate stage deals with conflicts of readings, such as the conflicted location information of a sensor. Finally, the Virtualize stage would clean incorrect readings by combining readings of multiple types of sensors of which readings are related to each other.

The main advantage of the pipelined based architecture, ESP, is that it is easier to integrate into data stream management systems that deploy a sequential query plan. However, in WSN applications, the nature of data streams is dynamic and cleaning data streams in a sequential set of modules in ESP without a dynamic justification might not be efficient. Moreover, the work proposed just only a framework for cleaning sensor data, without clarifying a specific technique to clean data in dynamic sensor environments.

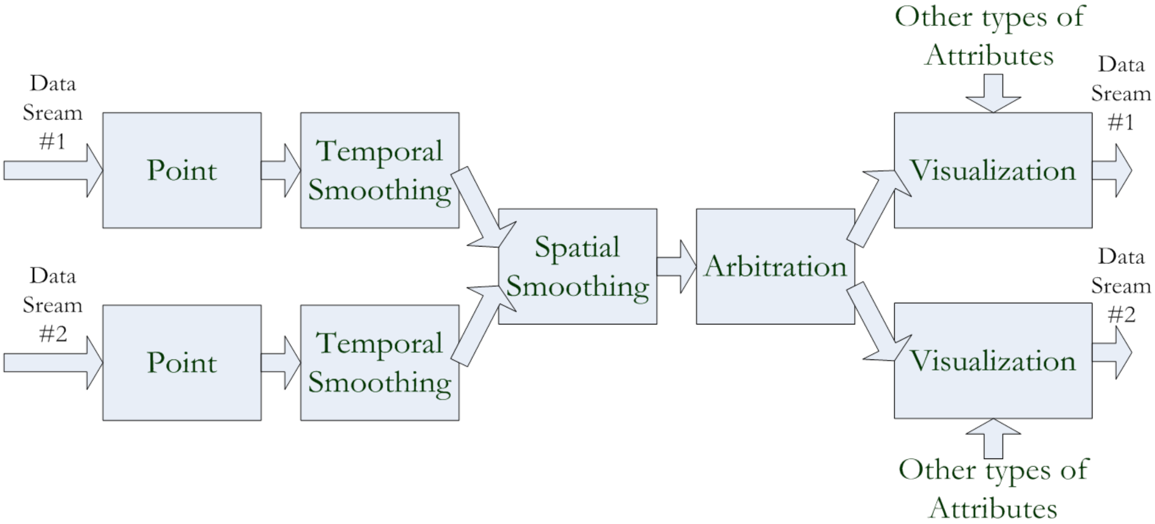


Figure 2.1 Modules in sequential based data cleaning method (ESP)

As the extension part of ESP framework, authors in [SJFW06] proposed a mechanism to estimate the quality of the cleaned data in streams for object detection applications. The authors proposed a quality track pipeline that will be used along with the cleaning pipeline in ESP to alleviate the drawbacks of the sequential ESP process in which the pipelined cleaning process cannot adapt to the dynamic nature of data in sensor applications. Two metrics, confidence and coverage, are used as measures of data quality. As designed for the object detection applications, the confidence metric accounts for false positive reports and the coverage metric accounts for false negative reports of the reality of object existence. The authors introduced formulas to calculate the confidence and coverage metrics in temporal and spatial smooth modules as well as the arbitration cleaning module.

The result of this work provides a reliability metric of the output of cleaned data streams. However, the design of this work is still restricted on the pipeline ESP data cleaning framework and it also restricts itself to the object detection applications. More importantly, it does not address how to correct any types of corrupted samples in data streams.

Later, authors in [JGF06] proposed a statistical Smoothing for Unreliable RFID data (SMURF) as an improvement of the ESP pipelined data cleaning framework for Radio Frequency Identification (RFID) applications. The key idea behind this work is that a RFID data stream could be viewed as a random sample of tags that are detected by RFID readers. Instead of using the fixed window size, the algorithm proposed in this work continuously computes and adapts a suitable window size based on the evaluation of binomial distribution of the observed readings. The algorithm employs the statistical sampling theory to clean the readings of single RFID tag and uses the Horvitz-Thompson estimator to clean an aggregated sample of a multi-tag population. SMURF is a mechanism to clean RFID data in the ESP pipelined data cleaning framework. Nonetheless, the SMURF cleaning mechanism was designed only for RFID systems. Data cleaning in an RFID system could not always be used in a WSN application because the RFID system mostly has a priori information of the observed data, such as range of tag values, class of tags and number of digits in a tag. Such data are discrete numbers, which are contrast to continuous numbers generated by wireless sensor devices. Cleaning the sensor data that contains multiple fields of continuous numbers intuitively needs a more complex technique.

2.1.2 Non-sequential based methods

In non-sequential based data cleaning methods, authors in [EN03] proposed a data cleaning method to detect and clean distance-based outliers by using the Bayesian theorem. The feature of sensor data and noise pattern are assumed to be a priori knowledge. In this work, the authors modeled the distribution of sensor data and noise by using Gaussian distribution with pre-defined and constant priori mean and variance values. This work was designed to answer a common set of user queries, such as Single Source Queries (SSQ), Set Non-Aggregate Queries (SNAQ), Summary Aggregate Queries (SAQ) and Exemplary Aggregate Queries (EAQ), with a "confidence level," which is the user-defined threshold that reflects the desired user's confidence of the expected query responses. However, this technique only considers the temporal relations of sensor readings from an individual sensor by using a priori, constant statistical knowledge, i.e., the distribution and related statistical parameters of the interested sensor readings are pre-defined and static. Considering that WSN data is normally dynamic, this technique is not suitable for typical WSN scenarios.

There are works deploying machine learning methods to solve the problem of inaccurately received sensor data. In [PS07], authors applied the neuro-fuzzy regression approach to estimate a new proper value for a sample with noise and a missing data sample. This approach formulates a regression model, which is derived from a training sample set of sensor readings. This model relies on the data feature of the entire sensor network and the spatio-temporal relationships between sensors. The authors also addressed how to deal with the uncertainty of data readings by using a neural network based fuzzy logic system to tune the parameters and structure of the regression prediction system. Another work that uses a technique in machine learning is addressed in [ZCWL07]. The authors proposed to clean the sensor data by using the moving-average based method to predict the corrupted or missing sensor readings as to address the dynamic data features of sensor readings.

In [BPM09], authors introduced the belief-based non-sequential data cleaning framework with the concept of time of arrival to identify the unreliable data streams received at the base station. The concept of time of arrival is that the higher jitter of the received samples of a stream implies the lower and less inconsistent quality of the wireless transmission links along the route used to forward the data samples.

The authors developed a data cleaning framework called Time Of Arrival for Data cleaning (TOAD), as shown in Fig. 2.2, that addresses the dynamic spatio-temporal correlations between two static sensors that are selected from a static set of immobile sensors. The framework also considers the inconsistency of time when samples of a sensor data stream are received at the base station. With both the spatio-temporal correlations and the time inconsistency factor, the authors proposed a belief based mechanism to filter out any anomalies in sensor data.

At the base station, for each sensor node, TOAD pre-defines a list of neighboring sensor nodes that are believed to facilitate data cleaning. TOAD computes the spatial correlations among sensor nodes and updates the belief parameter for each sensor node in an online fashion. TOAD introduced three adaptive filter based smoothing modules--temporal filter, averaging and tap exchange--for cleaning a dirty sensor sample. When a data sample of a sensor needs to be cleaned, TOAD justifies the most suitable smoothing module from the highest belief parameter that the corresponding sensor has with the pre-defined neighboring sensors. This work directly overcomes limitations of the ESP pipelined data cleaning framework in that it justifies the most suitable smoothing method according to the correlation and reliability between the neighboring sensors' data streams. However, for a sensor node, this framework assumes the static list of neighboring sensors that are expected to help clean data of the corresponding node. This assumption will not be applicable to the dynamic network topology of mobile sensors in mWSNs.

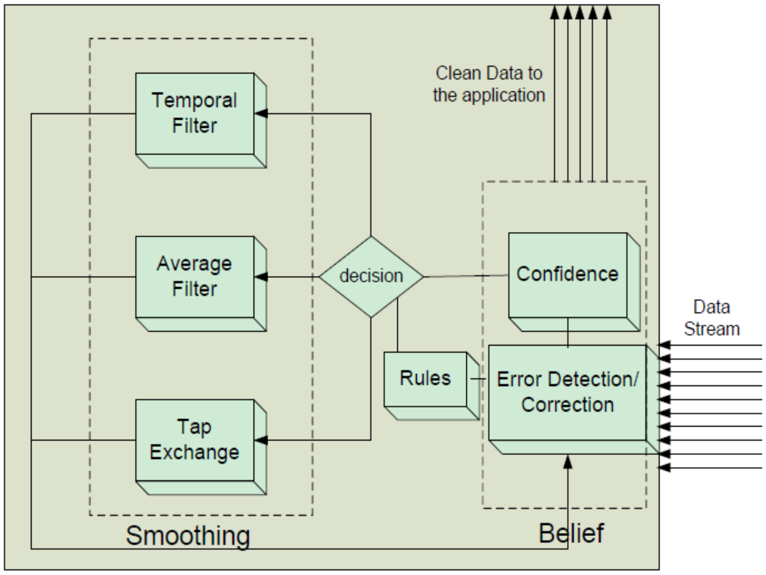


Figure 2.2 Time of arrival for data cleaning technique [BPM09]

Including the research milestones mentioned above, a number of different approaches to clean sensor streams in static WSNs have been proposed. However, they are all restricted in that the deployed sensors are assumed to be static and the contextual relationships in time and space among sensors remain unchanged. Since sensor mobility creates dynamic contextual relationships among sensors, the existing data cleaning methods for sensor networks cannot be applied to mobile environments.

2.2 Data Cleaning in Mobile Wireless Sensor Networks

Unlike data cleaning methodologies for static WSNs, data cleaning in mWSNs needs to consider the dynamism of sensor behaviors and relationships among themselves due to sensors' mobility. As there is no dedicated research work attempting to clean sensor data in mobile environments, we start reviewing the literature on the key task in mWSN data cleaning methodology, which is the identification of dynamic properties of sensor trajectories.

Normally, mobile sensors would periodically report their readings attached with the corresponding localization output. The sequence of corresponding sensor location and time represents the trajectory data of each sensor. Research related to trajectory data can be categorized into two main classes, offline and online. The offline class is a group of research works of which methodologies prepare, process, and extract information out of trajectory data, which are statically stored in databases or a data warehouse. Research in this class includes, but is not limited to, trajectory similarity search [FGT07] and trajectory pattern mining [GNPP07], etc. In contrast, research in the online class addresses the methodology that needs to manipulate a high volume of streaming data. The applications relying on the online class methodology typically need a real-time analysis for promptly taking a suitable action.

Since this research focuses on an online methodology, we here mainly review related works in the online fashion. In trajectory and moving object related works, there are attempts to resolve specific problems directed only to trajectory data. They do not address related problems that we address in this research. The example research works mentioned here include location prediction [MPTG09], trajectory classification [LHLG08], trajectory reduction [GTW$^+$10], density query [JLOC06] and enhanced semantic annotation [YCP$^+$11,Yan09]. Although some works [TTD$^+$09,TPL$^+$10,Tra11] address the manipulation of the uncertainty of trajectory data streams, they do not associate their solutions with sensor data. The research, which is the most relevant to data cleaning, addresses outlier detection [BCFL09,LHL08,LHKG07] from the trajectory perspective in which the proposed solutions can detect a different trajectory pattern of a moving node compared to those of its counterparts. In sum, we found none of them proposed a method in which the relationships between sensor trajectories, contexts and readings are coordinated to clean data streams in mWSN systems.

CHAPTER 3

VIRTUAL SENSOR FOR MOBILE SENSOR DATA CLEANING

Missing data is a common data uncertainty occurring in mobile wireless sensor networks. Node mobility prevents existing data cleaning techniques designed for stationary counterpart to properly estimate the missing data. In this chapter, we present a novel method to clean the missing data in mobile sensor environments. This method employs computation and memory resources at the base station to establish virtual static sensors collaborating with traditional adaptive filters in prediction model. With certain constraints given below, implementing the virtual sensor based algorithm cleans more than 80% of missing data, an increase in performance as compared to existing cleaning methods without any additional hardware implementations.

3.1 Introduction

In chapter 2, we have reviewed literature related to methods in data cleaning. All previous works have not dealt with the mobility of sensors which is an irrelevant issue in stationary WSN. In addition, there are attempts in literature that used the term of "virtual sensor," to resolve problems in WSN. Researchers [KPJ06,KSR08] deployed virtual sensors as a middleware layer to collect data from various heterogeneous stationary WSN. Jayasumana et al. [JHI07] proposed virtual sensors to collaborate for an efficient resource utilization protocol that supports network operations and maintenance over a network. Although the term of virtual sensor is not newly used for WSN, a proper algorithm to clean sensor data using virtual sensors has never been addressed.

In this chapter, we deploy the concept of virtual sensors to help cleaning data for mobile wireless sensor networks. We designed an architecture combining a novel online method using the concept of Virtual static Sensor (VS) and the Normalized Least Mean Square (NLMS) adaptive filter based predictive model [SH05] to address cleaning the missing sensor data in a mobile environment of sensor networks.

3.2 Assumptions and Methodology

We assume that the network operation is under single domain and based on a centralized architecture. Sensors move within a known and bounded area. All data successfully delivered to the base station is non-duplicated, noise-free and synchronized before proceeding to the following data cleaning process.

3.2.1 NLMS-based Linear Adaptive Filter

The adaptive filter is a tool in digital signal processing which can be applied to signal/data prediction when the nature of the data is time-variant; the statistical properties evolve over time. The adaptive filter is used in a prediction model as shown in Figure 3.1. The adaptive filter, at each step *k*, takes the latest N samples of input *x[k]* to compute an output $y[k]$ as a prediction of desired value *d[k]* by, $y[k] = w[k]^T \cdot x[k]$, where *w[k]* is the filter weight vector and *x[k]* is the input vector, both with dimensions *Nx1*. The error signal *e[k]* is the scalar value of difference between the actual desired value *d[k]* and output signal $y[k]$ computed as $e[k] = d[k] - y[k]$. The error signal *e[k]* feedbacks to the adaptive filter and updates all elements in the weight vector *w[k]* to, typically, minimize the mean square error. Among variety of adaptive filter techniques, the NLMS method is chosen due to its low computational complexity and its adaptive convergence rate to suit the evolving statistical properties of the nature of mWSN data. The NLMS algorithm applies the error signal to update the weight vector as follows: $w[k+1] = w[k] + \frac{\alpha}{x[k] \cdot x^T[k]} e[k] \cdot x[k]$, where α is a constant value between 0 and 1 [SH05].

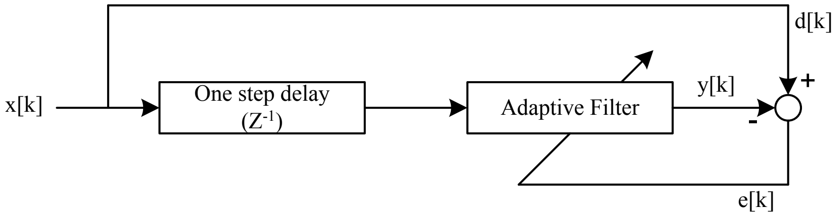


Figure 3.1 Adaptive filter in prediction model

3.2.2 Virtual Static Sensor

A Virtual static Sensor (VS) is, in fact, a memory space allocated at the base station to temporally store a vector value calculated from sensor data streams to represent a desired measurement from a specific location. In the aspect of online data processing, the data stream that is updated to a VS is analogous to a data stream received from a real sensor and temporarily stored at the base station.

To clean data in an area of interest, multiple VSs would be deployed. Before deploying VSs for the cleaning process, there are two components to be defined by system administrators which are the location and coverage area of each VS. The location of VS will be used to calculate the data for updating to a VS. The coverage area is needed for: (1) it is the area where each VS can obtain the data from the visited sensors and (2) it is the area where a VS can help other real sensors clean their missing data.

After defining locations and coverage areas of all VSs, the base station will update the data from real sensors, which enter the coverage area of a VS, to the corresponding allocated memory space. As mentioned, the VSs are not physically implemented and directly sensed the desired measurement in the area of interest. A method to update the VS data needs to be defined. To update data of a VS, we assume that the closer the real moving sensor is located to the corresponding VS, the more similar the data is from that real sensor to the data measured at the VS location. Therefore, at a specific time, we update the data for each VS by calculating the weighted average value of data indirectly proportional to the distance between each traversing sensor and the corresponding VS. The details of how a VS collects/updates its data are explained in the next section.

Because each VS does not move and is updated with the data based on the measurements physically sensed by all moving sensors in its coverage area, the VS knows best, compared to moving sensors, about the measurement trend in its nearby area. Therefore, the VS is helpful for the cleaning process to estimate missing data of real sensors in its vicinity. As an analogy, a VS acts as a host who knows best about events happening around her home and can provide, if needed, information around her home to her visitors.

To achieve an efficient performance of data cleaning in real implementations, the coverage area and location of a VS must be well defined. The system administrators who operate the data cleaning process can design the layout of each VS coverage area. Each coverage area is a sub-area where the desired measurements are expected to be similar -- the variance of measurements should not be large. For example, the system administrators who know the layout of a shopping mall should not place a VS in a location with a coverage area that includes both indoor and outdoor areas. This is due to their incompatible data features such as unequal mean value and high variance. That is, the more precisely the system administrators can identify the boundary of such dissimilarity of data features, the higher the cleaning performance they can achieve.

In cases where the system administrators might not precisely know about data feature differences in the area, the cleaning performance is likely to be lower. To gain higher performance in such situations, a smaller VS coverage area is preferred. This situation creates a tradeoff between the accuracy and the amount of the cleansed data. However, the problem we are addressing is not focused on such environments.

3.2.3 Adaptive Filter-based Data Cleaning

With the combination of adaptive filter and VS concepts, our data cleaning approach consists of three main steps: (1) VS data update, (2) VS data prediction and (3) real sensor data cleaning.

1) VS data update

To update the data to the memory space assigned for each VS at a specific time, the base station will collect data of all sensors which pass by the coverage area of a VS and then calculate the desired value for the VS by weighted averaging the collected data. The weight is indirectly proportional to the distance between the VS and sensors traversing in the VS area. We assume that the closer the real sensors are to the location of corresponding VS, the more weight their data are given in data estimation to update the VS data. The data for the VS can be calculated as follows:

$$S_{vs} = \frac{\sum_{i=1}^{n} S_i \left(\frac{1}{\left\| l_i - l_{vs} \right\|} \right)}{\sum_{i=1}^{n} \left(\frac{1}{\left\| l_i - l_{vs} \right\|} \right)}$$

,where

S_{vs} : The data that updated to the corresponding VS.

S_{i} : The data sensed by the real sensor *i* in the coverage area of corresponding VS.

n : The number of sensors in the coverage area of corresponding VS.

l_{vs} : The pre-defined static location of the VS.

l_i : The location of real sensor *i.*

The S_{vs} value will be updated to the VS as the current data sensed by the VS. The algorithm updating the data to a VS is illustrated as algorithm in table 3.1.

2) VS data prediction

Due to the unpredictable movement of real sensors in the network, it is likely that a VS may not have sensors passing by to update data, which parallels to data missing in a real sensor. This uncertainty of updates reduces the potential of a robust cleaning process. To maintain a stream of updates in each VS, there is an NLMS adaptive filter running in the prediction model at the base station to temporally estimate the missing VS data. However, without real updated data, the accuracy of the data estimation gradually deteriorates and the predicted data after N (the size of filter weight vector) consecutive predictions will not be used. Once there are N+1 consecutive missing data samples from the VS, it stops predicting any missing data and does not restart predicting there are at least N consecutive data updates received from real sensors.

Table 3.1 VS data update

```
// Input:  1) Location of n sensors at time t: $l_t = \{l_{1t}, l_{2t}, l_{3t}, ..., l_{nt}\}$
//         2) Data sample of sensors at time t: $S_t = \{s_{1t}, s_{2t}, s_{3t}, ..., s_{nt}\}$
//         3) Location of k VSs: $l_{vs} = \{l_{vs1}, l_{vs2}, l_{vs3}, ..., l_{vsk}\}$
// Output: The k data to be updated to the k VS by weighted averaging based on the
//distance from each sensor to the corresponding VS at time t: $Y_t = \{y_{1t}, y_{2t}, y_{3t}, ..., y_{kt}\}$
Procedure VS_UPDATE_ALGORITHM ($S_t$, $l_t$, $l_{vs}$)   //At time t
    for $VS$ = 1 to k do // Loop the number of VSs
        for $RS$ = 1 to n do // Loop the number of real sensors
            if ( ($l_{RSt}$ is in the coverage area of the corresponding VS)
              && ($s_{RSt} \neq$ missing data) ) then
                //Compute the accumulated total distance
                $D_t = D_t + 1/||l_{RSt}–l_{vs}||$
                $A_t = A_t + ( S_{RSt} / ||l_{RSt}–l_{vs}|| )$
                end
            else
                skip to the next sensor;
            end
        end
        // Compute the updated value to the VS
        $Y_{vst} = A_t/D_t$
    end
end procedure
```

3) Real sensor data cleaning

If the base station detects missing data in a real sensor, it starts cleaning the missing data in the stream. Normally the sensed data, at a time instance, is sent along with location information of the sensor in the same packet. When a sample of data is missing, it is common that the location information is missing as well. Since our cleaning method selects the most suitable VS to clean the missing data based on the location of the real sensor, an arbitrary location prediction algorithm [MZM+09,MPTG09] is required to locate the sensor when the missing data occurs. However, if the location information is delivered to the base station out-of-band to that of the sensor data and the location of the sensor is available, the location prediction is not needed. This case is similar to when a selected location prediction can perfectly locate the sensor. Once the real sensor is located and the most helpful VS is selected, the cleaning process will replace the missing data of the real sensor with the data of the selected VS at the corresponding time instance.

3.3 Evaluation and Analysis

We simulated an mWSN environment using the software package MATLAB. We compared the efficiency of our method to method in [BPM09] based on varied sensor densities, average sensor speeds and amount of missing data.

In this scenario, there are *n* real sensors moving randomly and collecting temperature data. The area of interest is 54m x 54m divided into 9 sub-areas of 18m x 18m each. There are three different sub-area types with different average temperatures: (1) Indoor area, (2) Shaded outdoor area and (3) Outdoor area.

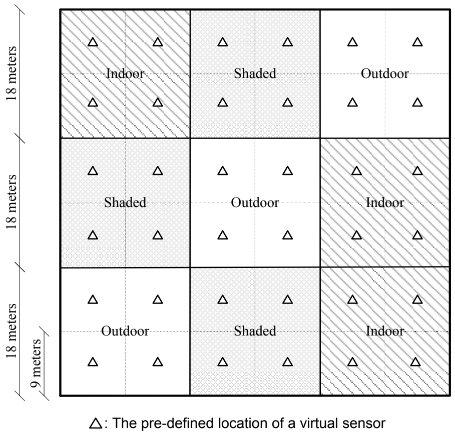


Fig. 3.2 Layout of tested area

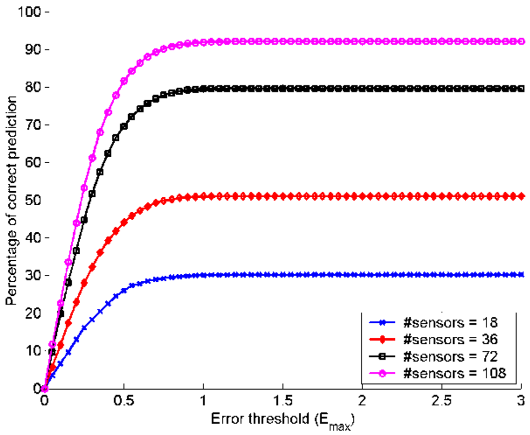


Fig. 3.3 Percentage of correctly cleaned data under each error threshold varying with the number of real sensors.

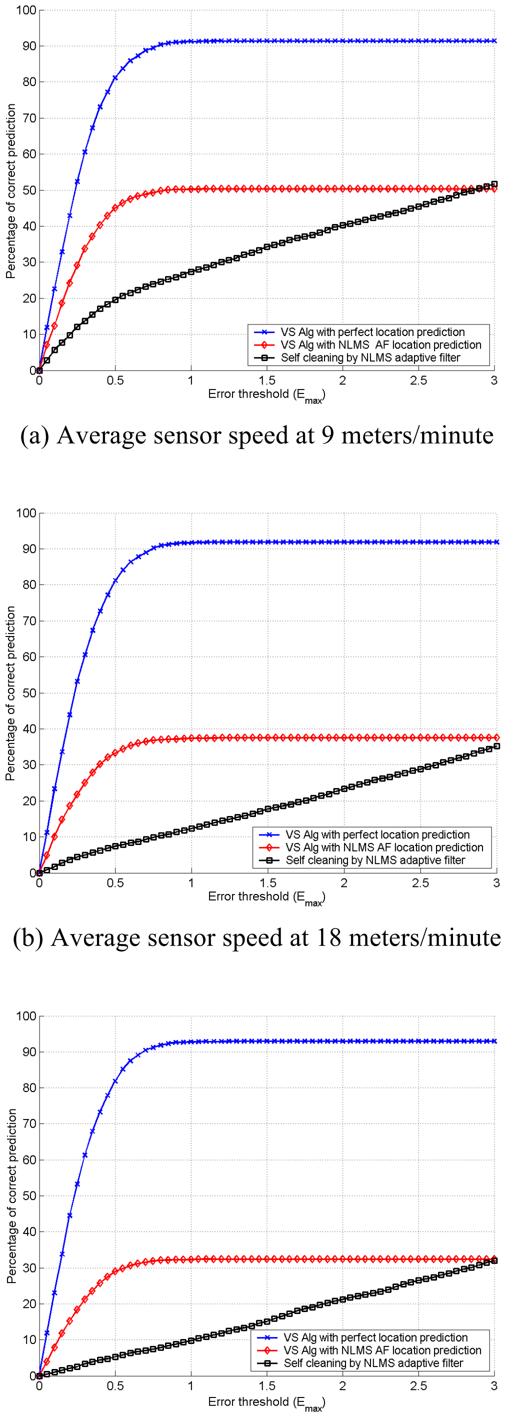
Each area category contains temperature values based on a normal distribution with mean values µ1, µ2 or µ3, respectively, and fixed standard deviation of 0.5ºC. The data set is from an hourly observation table at the Asheville regional airport during January 1-15, 2007 [Nat07]. All mean values evolve over time with µ3 approximately 6ºC higher than µ2 and 13ºC higher than µ1. We assume that the system administrators allocate 36 VSs to evenly cover the whole area of interest without overlapping coverage areas. The layout of the area is depicted as in Figure 3.2.

In each simulation, we assume that each moving sensor samples the data every 30 seconds. We simulated 40000 data samples per each real sensor and used it as a referenced data set. To generate missing data in the set, we randomly took out a set percentage of samples. The adaptive filter running in each VS has a length of N = 5 taps. The percent ratio of the amount of correctly estimated missing data to the total missing data samples indicates the effectiveness of the algorithm. Data is counted as "clean" only when the absolute difference between the estimated data and the referenced data is lower than a pre-defined error threshold, E_{max}, which is the user-defined error margin indicating whether the data estimation is correct.

The evaluation was performed by varying three parameters: (1) number of real sensors in the system, (2) average sensor speed and (3) the amount of missing data. First, we varied the number of real sensors in the area of interest but maintained the average sensor speed at 9 meters per minute with 30% missing data. As shown in Figure 3.3, the result shows that the increasing the number of real sensors in the environment, the better the performance of our algorithm. The increasing number of real sensors in the area increases the rate of data updated to each VS. The higher rate of data updates to VS lowers the amount of missing data in real sensors that cannot be estimated. By considering at the E_{max} of 0.5ºC, our algorithm can correctly clean more than 80% of the missing data when there are 108 moving sensors to 36 deployed VSs; 3 times the number of deployed VSs. On the other hand, this result also indicates that, to maintain a specific level of cleaning performance, the number of deployed VSs must be limited relative to the number of real sensors operating in the area. However, in order to maintain the VS coverage over the whole area of interest, when there is a smaller number of VSs, the larger the VS coverage is required. Whenever a VS coverage area is expanded and covers the areas with different categories, the cleaning accuracy would be decreased. Therefore, this consequence proves that the virtual sensor based algorithm works without compromise in the cleaning performance only when the sensor density in the area of interest is high. The required number of real sensors is dependent on the required level of the guaranteed performance.

Next, Figure 3.4 shows that our method is unaffected by different sensor speeds when the location information field is not missing or perfectly retrieved by a location prediction algorithm. With the number of real sensors fixed at 108 and the average sensor speed varied, 80% of the missing data is cleaned with an E_{max} = 0.5ºC. Meanwhile, when an existing imperfect location prediction algorithm like NLMS adaptive filter [SR06] is applied, the performance decreased. This is the result of the performance degradation of location prediction algorithm that provides more incorrect location information to the cleaning process when the sensor speed is faster. Although the performance is dropped when applied the imperfect location prediction, it still approximately 20% higher than that of temporal cleaning method [BPM09].

As the number of missing data increases, less data get updated in the VS. This reduces the amount of available VS data that could be used in the data cleaning process. Therefore, as shown in Figure 3.5, when the amount of missing data increases the performance of our algorithm deteriorates. Nevertheless the performance is still higher than that of the temporal self-cleaning method [BPM09].

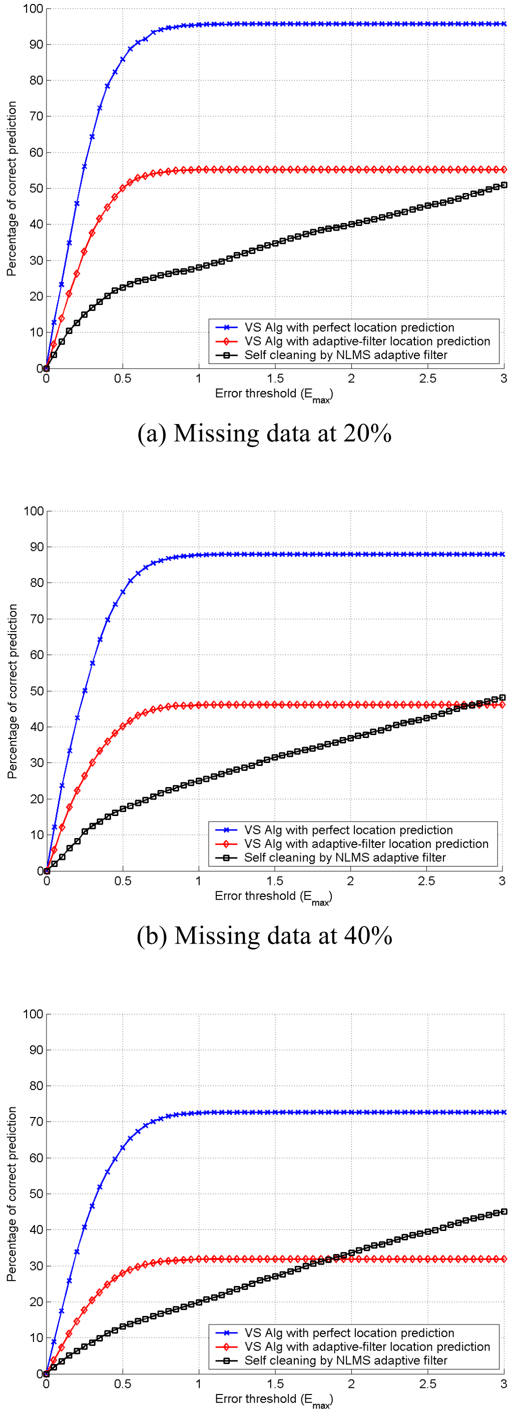


(a) Average sensor speed at 9 meters/minute

(b) Average sensor speed at 18 meters/minute

(c) Average sensor speed at 27 meters/minute

Fig. 3.4 Percentage of correctly cleaned data at 30% missing data with various sensor speeds among (1) VS algorithm with the perfect location prediction (2) VS algorithm with NLMS adaptive filter based location prediction, and (3) Self temporal cleaning by using NLMS based adaptive filter



(a) Missing data at 20%

(b) Missing data at 40%

(c) Missing data at 60%

Fig. 3.5 Percentage of correctly cleaned data at 9 meters/minute sensor speed with various amount of missing data among (1) VS algorithm with the perfect location prediction, (2) VS algorithm with NLMS adaptive filter based location prediction, and (3) Self temporal cleaning by using NLMS based adaptive filter

3.4 Summary

In this chapter, we present a novel method to clean missing data in mWSN applications. This method combines virtual sensors with an NLMS adaptive filter. Results demonstrated that, when the location information of the sensor is not missing or fully retrieved by a location prediction, our method cleans more than 80% of missing data, is independent of the sensor speed when the number of moving sensors is at least 3 times the number of VSs. Since the method does not need any additional hardware implementations, it suits to mWSN applications where the area of interest is temporary for operations.

CHAPTER 4

BELIEF-BASED CLEANING IN TRAJECTORY SENSOR STREAMS

The imprecision in data streams received at the base station is common in mobile wireless sensor networks. The movement of sensors leads to dynamic spatio-temporal relationships among sensors and invalidates the data cleaning techniques designed for stationary networks. As one of the first methods designed for mobile environments, we present a novel online method to clean the imprecise or dirty data in mobile wireless sensor networks. Our method deploys a belief parameter to select the helpful neighboring sensors to clean data. The belief parameter is based on sensor trajectories and the consistency of their streaming data correctly received at the base station. The evaluation over multiple mobility models shows that the following method outperforms the existing data cleaning algorithms, especially in sparse environments where the node density in the system is low.

4.1 Introduction

In chapter 3, it has been observed that the cleaning method based on the concept of virtual sensor does not consider the non-synchronization of sampling time among sensors, and its performance is limited by the node density in the system. Thus, we are motivated to clean mWSN sensor data with an online method to satisfy the real-time applications. The contributions in this chapter are:

- We introduce a belief-based sensor selection method to identify the group of sensors that is helpful in cleaning data based on their current trajectories and the quality of their data streams.

- We present a novel online data cleaning method designed for the dynamic environment in mWSN applications. Our evaluation results show that the cleaning performance of our method outperforms those of virtual sensor-based method in [PP10] and a method designed for stationary sensor networks in [ZCWL07].

4.2 Problem Statement and Assumptions

Assuming that there is a pre-process operating to detect the dirty data, such as outliers, non-ordered data sequence, out-of-date data and missing data, etc., such dirty data is discarded by the system. We develop an online algorithm to clean the dirty data streams in mWSN environments. The designed cleaning process is centralized based architecture, i.e., all cleaning mechanisms including the detection of dirty data and data stream management are conducted at the base station where all sensor data streams are forwarded.

In practice, the trajectory data expressing the time-location information and the sensor measurements from a sensor could be delivered to the base station via the different channel as an out-of-bound transmission. Although the sensor measurements are dirty and need to be cleaned, we assume that the trajectory data is correctly received at the base station.

We focus on cleaning the dirty data from sensors, which are moving in a pre-defined area of interest. We assume that multiple sub-areas form up the area of interest. The level of reading in the same sub-area is similar and different from that of adjacent sub-areas. The boundaries among sub-areas are also assumed to be known.

4.3 Belief-based Cleaning Method

This presented data cleaning method is an area-based approach assuming a priori knowledge of sub-area boundaries. The cleaning process computes the replacement of dirty data by utilizing the readings from a group of sensors that are believed to be offering enough reliable readings from a specific sub-area. In this section, we explain the developed cleaning method in detail. We first discuss how a group of neighboring sensors is selected for collaborating in the cleaning process. We then describe how the dirty sample is cleansed based on the distance function in both time and location of sensors.

4.3.1 Belief-based Sensor Selection

With the number of deployed sensors in practice, brute-force methods to select the most correlated data readings are not practical. Based on a priori knowledge of sub-area boundaries, each sub-area has been indexed and matched with a belief table. This approach is using the belief table, which contains the updated belief degree of each sensor for each sub-area. For a sub-area, the belief degree of each sensor represents how trustworthy a sensor could help cleaning the dirty readings measured within the sub-area at a specific time. It is based on two parameters, which are (1) alibi degree and (2) detection rate of dirty data, explained as follows:

1) Alibi Degree (A)

The alibi degree is computed at a specific time to show the accommodation level that a sensor experiences and reads the dedicated measures within a sub-area. At a specific time, the higher the alibi degree of a sensor, the more the sensor operates within the corresponding sub-area. The alibi degree is computed from residence vector and the frequency of existence in the sub-area.

The residence vector expresses a series of existence of a sensor located in a sub-area. The sensor existence in each sub-area is computed from the trajectory data of each sensor received by the base station. The members of the vector are stored in the allocated window of memory space. They are of Boolean type; 1 when the sensor is located within the corresponding sub-area and 0 when the sensor stays outside that sub-area. For example, illustrated in Fig. 1, a sensor is traversing across a sub-area. If the allocated window size equals 9, the residence vector from time sequence t_1 to t_9 will be [0 1 1 0 1 1 0 0 0].

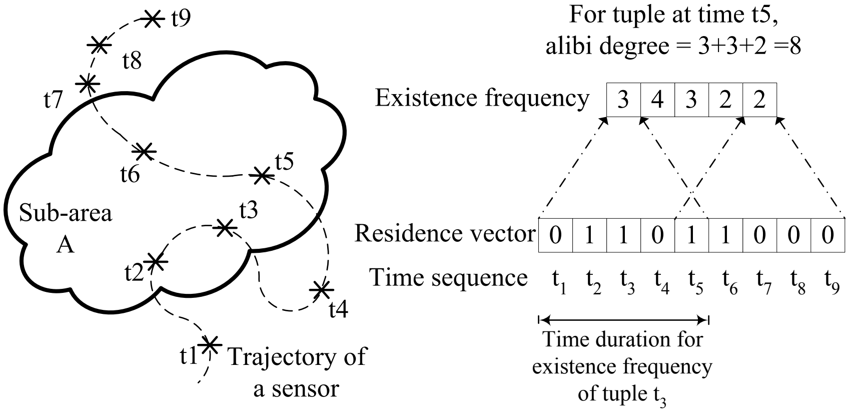


Figure 4.1 An example of alibi degree calculation

While the residence vector is updated, the frequency vector is also computed and stored in another window of memory. For a sensor, each member of the frequency vector represents the frequency of the sensor existence within the corresponding sub-area. The frequency of the sensor existence is calculated per time duration. For instance, Fig. 1 shows that the time duration for calculating the frequency of existence is set to 5 samples.

The existence frequency at time instance t_3 is the sum of members of residence vector from instance t_1 to t_5; that of tuple t_4 is sum of members of residence vector from instance t_2 to t_6, and so on. Note that existence frequency at the time sequence t_3 contains residence information in the following tuples, which are those of t_4 and t_5. As it would be later explained, the existence frequency at time instance t_3 is needed to clean a dirty sample at sequence t_3. The cleaning process for the instance at t_3 would be delayed by half of the user-defined length of time duration. The higher frequency value implies a greater chance of the sensor having experience within the corresponding sub-area.

With the allocated window size of 9 and time duration of 5 tuples as shown in Fig. 1, the existence frequency vector would be fulfilled after the trajectory data of tuple t_9 is received by the base station. The alibi degree can then be computed as a dot product of residence vector and frequency vector. The maximum value of alibi degree is equal to length of existence frequency vector times its time duration in samples. At the time of tuple t_5, the alibi degree would then be equal to 3+3+2 =8, and equals 8/25 after normalized.

2) Detection Rate of Dirty Data (D)

Although two sensors are in the same sub-area, their different trajectories can lead to different environments affecting the quality of data delivery. Here, we present the detection rate of dirty data to inversely represent the reliability of the data stream of each sensor. As the area-based parameter, the detection rate of dirty data shows the quality of streaming data received from a sensor residing in the corresponding sub-area. As an online method, we introduce the calculation of the detection rate of dirty data as a ratio of the cumulative number of detected dirty samples to the number of all samples that the sensor measured within the corresponding sub-area.

Note that we assume that a pre-processing module to detect the dirty data exists and correctly detects the corrupted samples. Intuitively, the lower the detection rate of dirty data, the more reliable the data stream of the sensor residing in a particular sub-area.

3) Belief Degree and Sensor Selection

At a time instance, the belief degree of each sensor will be calculated and updated to the belief table specific each sub-area. The belief degree would be increasing due to the alibi degree but decreasing due to the detection rate of dirty data. The derivation could be shown as in Equation 4.1. The high-level description in updating the belief table of all sub-areas is illustrated in Table I.

$$\beta = (\alpha \cdot A_N) + (1 - \alpha) \cdot \ (1 - D) \qquad (4.1)$$

where β : belief degree

α : belief coefficient

A_N : normalized alibi degree

D : detection rate of dirty data

Our approach to clean a corrupted sample utilizes the readings from sensors, which are reliable enough. The sensors with the β value higher than a belief threshold (β_{th}) would then be selected to collaborate in the cleaning process. The proper values of belief threshold (β_{th}) and belief coefficient (α), ranging between 0 and 1, are depending on applications and the nature of measurements of the system. For example, if the performance of the dirty data detection module offers a large uncertainty, the belief coefficient would be set close to 1.

4.3.2 Belief-based Cleaning Process

After a group of sensors is selected to help cleaning the dirty data for the target sensor, a cleaning process will compute the cleansed value to replace the value of the dirty sample. The calculation of cleansed data considers (1) the time difference between the time that each available data of the selected sensors are sampled and the time when the target sensor senses the dirty sample, and (2) the distance between the selected sensors and the target sensor when the target sensor senses that dirty sample. Only readings of the selected sensors sensed in the same sub-area where the dirty data is measured are eligible to be deployed in this belief-based cleaning process.

We assume that the lower the sampling time difference and the location distance between the selected sensors and the target sensor, the more similar the data from selected sensors would be to the actual measure of the target sensor. We here use that a cleansed value will be equal to a weighted average that is indirect to a distance function in sampling time and location, as shown in Equation 4.2.

$$d_c = \frac{\sum_{i=1}^{k} d_i \cdot \frac{1}{\Delta t(d_d,d_i)} \cdot \frac{1}{\Delta L(d_d,d_i)}}{\sum_{i=1}^{k} \frac{1}{\Delta t(d_d,d_i)} \cdot \frac{1}{\Delta L(d_d,d_i)}} \tag{4.2}$$

Where k : The number of data samples of selected sensors residing in the sub-area

d_c : Cleansed data of the target sensor

d_i : The eligible data from selected sensors

$\Delta t(d_d ,d_i)$: Difference in sampling time of the dirty sample d_d and the eligible data d_i

$\Delta L(d_d,d_i)$: Location distance of target sensor and selected sensors when the target sensor senses the dirty sample d_d

4.4 Evaluation and Analysis

In this section, we summarize our experiment analysis to evaluate the performance of the presented algorithm. To our best knowledge, the virtual sensor-based method (VS) in [PP10] and the belief-based method (BB) are the first algorithms attempting to clean dirty data in mWSN environments; therefore, the performance of the belief-based method will be compared with the VS method and another designed to clean data in static WSN based on the moving average method [ZCWL07].

The performance of algorithms is evaluated in a simulated scenario in which there are *n* sensors moving randomly and sensing the temperature data. The 200 x 200 m^2 area of interest is divided into 9 sub-areas, as shown in Figure 4.2. These 9 sub-areas will be classified into 3 categories based on the area characteristics: (1) Indoor area, (2) Shaded outdoor area and (3) Outdoor area.

Each category exposes temperature values based on a normal distribution with a different mean but the same standard deviation of 0.5ºC. The average temperature value of each category evolves by time according to the change of data trend collected from the Asheville Regional Airport, North Carolina, from January 1-15, 2007 [Nat07]. The mean temperature in indoor areas is roughly 7ºC lower than that of shaded area and 13ºC lower than that of outdoor areas.

In each round of simulation, each sensor randomly starts sensing data during 0-30th second, and it would constantly sample the data every 30 seconds. With a variety of node densities, each sensor senses 1200 samples as a referenced data set. As we assumed that the dirty samples are detected before progressing to the presented cleaning module, we randomly assigned a fixed percentage of all samples as the detected dirty data that need to be cleaned. The window size of the residence vector equals 9, and time duration for the existence frequency is set to 5.

We considered three mobility models – (1) random waypoint, (2) nomadic, and (3) random street in our evaluation. The random waypoint [JM96] is a classic mobility model that each node will move from its current location to a randomly selected new location with a random speed and it will pause before moving to another new location. Instead of the independent random movements, the nomadic mobility [CBD02] represents groups of sensors that collectively move from one location to another. This mobility suits to scenarios of, for example, a class of students touring in a museum. The random street [AS10] is a newly established mobility model that mimics scenarios when there are path constraints such as walls, buildings and motorways presented as in a real map.

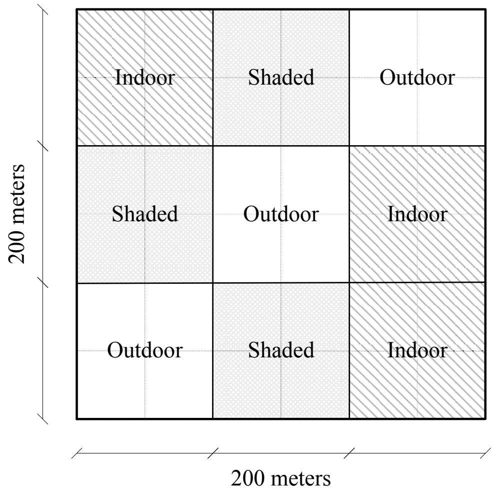


Figure 4.2 Layout of tested area of interest

Table 4.1 Belief table update

```
// Input:     The location data of sensors in window space at time $t_k$
// Output:    The updated belief table (T) of all sub-areas
// Update the belief table of each sub-area, one by one
Procedure Belief_update
for subA = 1 to S              // S is number of all sub-areas
  for i = 1 to N               // N is number of all deployed sensors
      Calculate the alibi degree;
      Calculate the detection rate of dirty data;
      Calculate the belief degree as shown in Equation (1) in this chapter;
   Update the belief degree matched with sensor(i)  in T;
  end
end
end procedure
```

We used the Bonnmotion mobility scenario generator [Bon02] to generate the trajectory data for all mobility models. In nomadic settings, the number of nodes per group is at 10 nodes with deviation of 2 nodes and the maximum group radius is at 15 meters. For the random street, we selected a real area with path constraints in Germany as defined in the GIS reference as the EPSG code: 31466; Gauss-Kruger zone 2. The maximum pause time is set at 60 seconds as similar to that in the random waypoint settings.

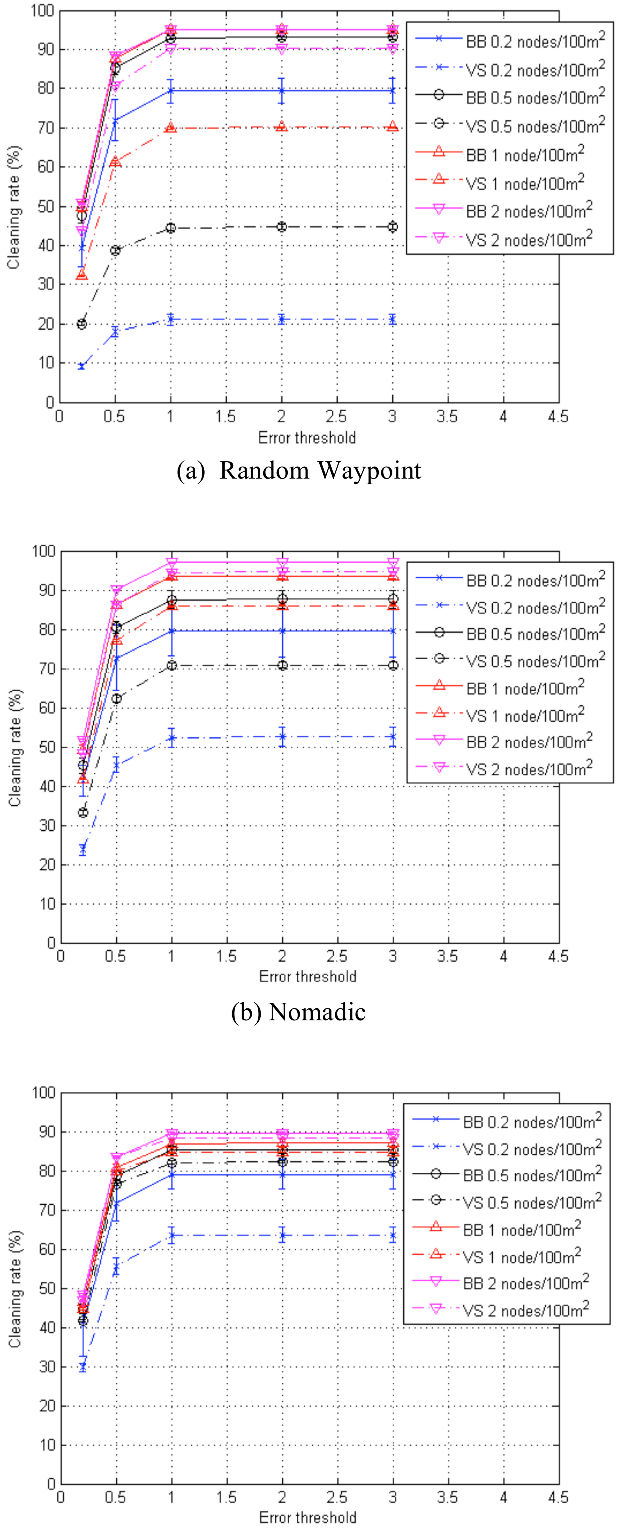
The performance of cleaning methods is evaluated by a ratio of the number of "successfully cleaned" samples to the number of whole detected dirty data. This ratio is referred to as the cleaning rate. A dirty sample would be successfully cleaned only when the absolute difference between the output of cleaning process and the referenced data is bounded under a user-defined error threshold.

For BB method, we experimented as the alibi degree and detection rate of dirty data are equally significant. We then set α equal to 0.5 and experiment with β_{th} at 0.7. As we assume that the effective average transmission range of a sensor node is around 20-25 meters, the coverage of VS is then set at 22.5 meters.

We first compared the cleaning performance with various densities of sensor nodes moving in the area of interest as shown in Figure 4.3. For all tested mobility models, the performance of our belief-based cleaning method is superior to that of the VS method especially when the node density is low. In random waypoint models with 0.2 nodes/100 m^2, the cleaning rate of the BB method exceeds that of the VS method for at least 50% at 0.5 error threshold.

We also evaluated the cleaning rate when the percentage of detected dirty data is varied as shown in Figure 4.4. For all mobility models, the cleaning rate of the belief-based cleaning method surpasses at least 25% compared to other tested methods.

Scenarios with different average node speed of 2 mph (human walking), 8 mph (biking) and 20 mph (car slowly running) were also experimented. The result in Figure 4.5 shows that the belief-based cleaning method outperforms the VS method for all mobility types. Although the cleaning rate of the belief-based method is degraded faster than the VS method in the nomadic mobility model, the superior performance is remaining up to the speed of car slowly running.

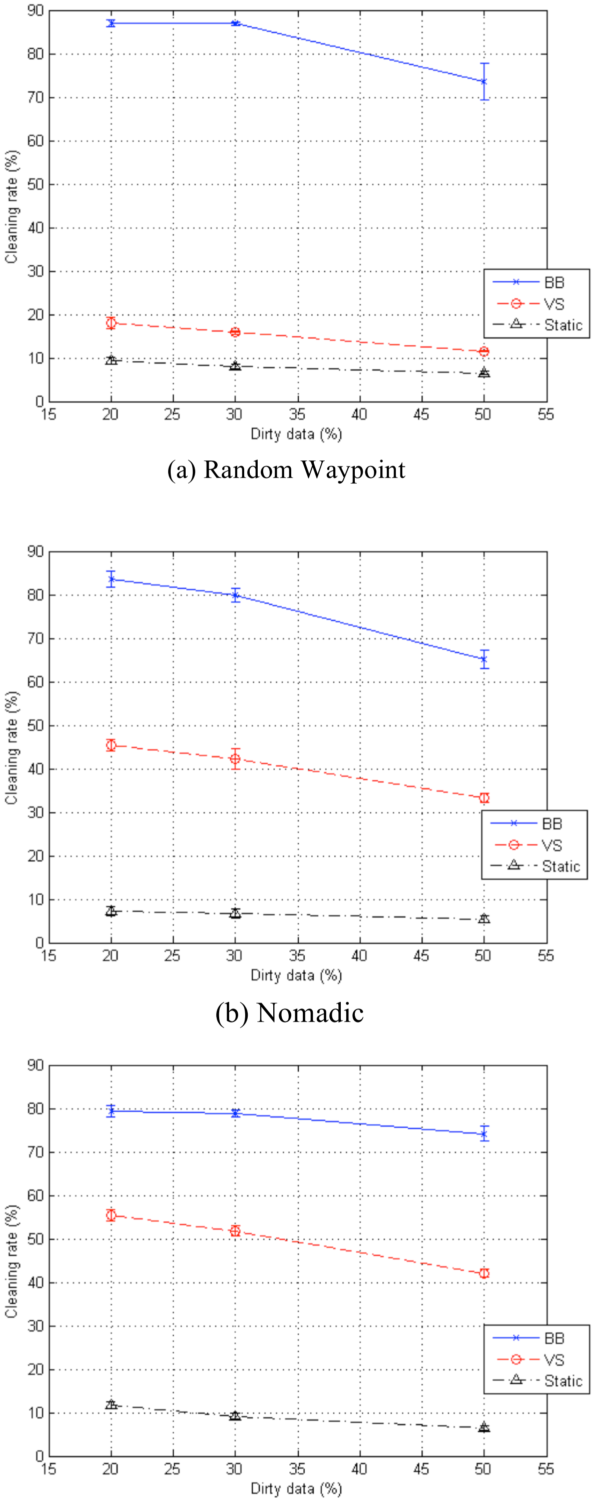


(a) Random Waypoint

(b) Nomadic

(c) Random Street

Figure 4.3 Cleaning performance with varying node density in different mobility models and dirty data of 20%

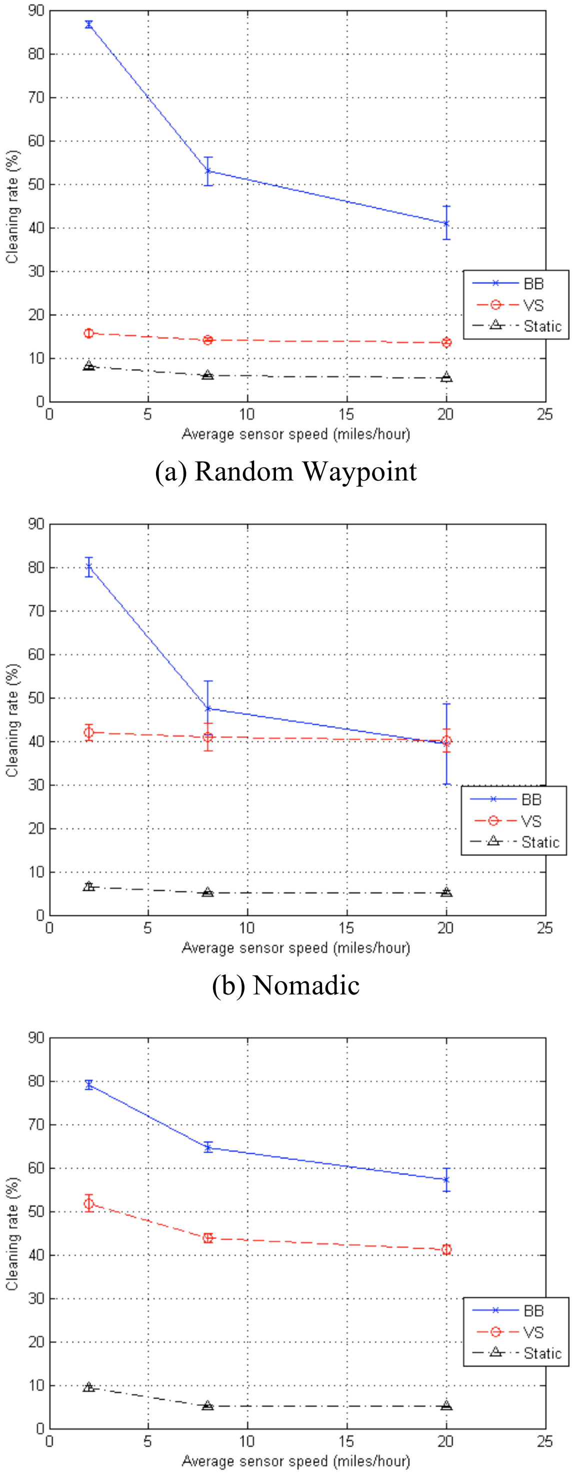


(a) Random Waypoint

(b) Nomadic

(c) Random Street

Figure 4.4 Cleaning performance with varying percentage of missing data in different mobility models at 2 mph average speed



(a) Random Waypoint

(b) Nomadic

(c) Random Street

Figure 4.5 Cleaning performance with varying average speed of sensors in the area in different mobility models and dirty data of 30%

4.5 Summary

In this chapter, we have presented a novel simple method of data cleaning suited to mWSN applications. Rather than relying on the static spatio-temporal relationships among sensors, which is invalid to mWSNs, we analyzed the area-based trajectory features as the residence pattern and existence frequency to reveal how a neighboring sensor can help in the cleaning process. Moreover, the cumulative detection rate of dirty data is also utilized to grade the trustworthy level of a data stream within per particular sub-area. The superior performance compared to that of the existing cleaning methods is demonstrated for various mobility models, dirty data rate and average sensor speed.

Since this work is one of the first solutions to clean dirty data in mWSN next to virtual sensor based method, there are more challenging limitations to overcome. The trajectory information can also be dirty or imprecise. Also, the area classification might be unknown and dynamic. One of the promising research directions is to find solutions to cope with such complex situations.

CHAPTER 5

SKETCH-BASED CLEANING IN SENSOR DATA STREAMS

As mentioned, the data imprecision received at a base station is common in mWSNs. In scenarios, data cleaning based on spatio-temporal relationships among sensors is not practical due to the unique, but commonly found, characteristics of sensor networks. The data cleaning method presented in this chapter deploys a sketch technique to periodically summarize *N* sensor samples into a fixed size array of memory and manages to recover values of missing or corrupted sensor samples at the base station. Our evaluation demonstrates that, with a small fixed portion of additional data transmission compared to original *N* data, this method outperforms the existing data cleaning methods, which assume the spatio-temporal relationships among sensors.

5.1 Introduction

Currently mobile wireless sensors have been deployed in ranges of applications [HBZ$^+$06,EGH$^+$08] mainly to monitor and collect a high volume of data streams. Meanwhile, a large amount of data transmitted by sensors is missing and corrupted at points of data collection [Int08]. For more than a decade, online data cleaning has been a research area, which is on focus of researchers in order to improve the quality of data streams of static WSNs. Common techniques are in forms of statistics, probabilistic, logics and machine learning methods. The majority of them deploys the temporal [EN03,PS07,ZCWL07] and spatio-temporal relationships [MSHR02,BPM09,DGMH04] of nearby sensors to help recover the missing or corrupted sensor data, assuming that data from these nearby sensors are correlated to each other.

Although this assumption is widespread and well used in static wireless sensor systems, there are common scenarios in which techniques that rely on this assumption cannot be applicable, for example:

(1) *Mobile wireless sensors:* In this case, the moving sensors could traverse through areas with significantly different environment, for example, temperature sensors that are moving indoors and outdoors. The data cleaning methods based on features of its own historical data of a sensor are not efficient. Furthermore, mobile sensors also create a dynamic network topology and transient spatio-temporal contexts among themselves. Correlations of data from a specific pair of mobile sensors are fluctuated and cannot be deployed to clean their data.

(2) *Sparse network:* Although tiny sensors typically have low-power transceiver units, micaZ sensors from the Crossbow, for example, can have a typical one-hop transmission range at 20-30 meters outdoors and up to 10-12 meters indoors based on our experiment. With this range of transmission, sensors may not measure a similar, shared environment, even when they are located next to each other or they are sharing a logical communication link.

(3) *Internal sensing:* There are many types of sensors that do not measure a surrounding phenomenon, for example, sensors measuring heart rate, blood pressure, and glucose level of patients or sensors gauging acceleration and gas level of vehicles, etc. The measurement from these types of sensors could change abruptly and not be likely to demonstrate a correlation to those of neighboring sensors.

We are motivated from these common scenarios where the existing online data cleaning techniques are not applicable. In this chapter, we present a sketch-based method that can clean data streams of sensor environments mentioned above.

The idea of the sketch technique was first introduced to determine a representative trend of data in time series [IKM00], which is the extension of a random projection technique [JL84]. This technique was designed to reduce the dimensionality of offline massive time series data. It is not satisfied in scenarios, where continuous querying and processing are required over the data stream [MSHM02]. Later, many different sketch techniques for data streams were proposed for various applications. They include item frequency tracking [CM05], data stream clustering [Ind03], page rank approximation [SBC+06] and entropy estimation [GL06] on data streams.

Our focus is to modify the Count-Min sketch method, which basically proposed a data structure to summarize an arbitrary set of data into a compact, fixed-size array that is usually small enough to fit within a cache [CM05]. The fixed array memory is reserved for counters to record the number of types of items in a data stream. However, the summary of data streams is presented as item counts and does not provide the temporal information of each item.

This method modifies how to update data into this fixed array to be able to store the sensor data with their relative temporal information, instead of only item counts without the corresponding temporal data. The new scheme of data update will allow the base station to recover the value of missing data or corrupted data, which have been prior detected.

The main contribution of this work is that we develop a sketch-based data cleaning method for data streams in mWSN environments, where moving sensors do not measure a shared phenomenon or are deployed in a sparse network.

5.2 Problem Statement and Assumptions

In this context, we consider sensor applications, where a number of moving sensors measure and deliver streams of their sensor data back to a centralized station. The imprecise and incomplete streams of sensor data received at the centralized station include, but are not limited to, outliers, missing and noisy data. However, we assume that there are arbitrary prior modules that function to detect such missing, imprecise and corrupted data samples. We assumed that all imprecise data samples are detected and then discarded. Thus, in this work, we will refer all imprecise samples as missing samples.

We focus on cleaning a numerical type of data from mobile sensors. We assume that sensors are periodically sampling with a pre-defined period before being deployed in the system. Each sample is paired with a sequence number to represent the order of sequential sampling. With a known sampling period, the sampling time of a missing data sample can be retrieved by referring to the sampling time of the adjacent sequence numbers.

5.3 Sketch-based cleaning

Principally, this cleaning mechanism consists of two main processes, the sketch and cleaning process. The sketch process is executed at sensors and the cleaning is processed at the base station. Each sensor will operate and transmit the outcome of sketch process periodically. The cleaning mechanism at the base station is then a one-to-one process. That is, it would deploy only the outcome of the sketch process, including recently received data of a sensor to estimate the value of missing data of the corresponding sensor.

Sketch process: Before the sketch process starts, each sensor will reserve two fixed arrays of memory of width w and depth d. One is called sketch array, $[a(1,1),\ldots,a(d,w)]$ and the other is the counter array that counts a number of updates of each corresponding member of the sketch array, $[c(1,1),\ldots,c(d,w)]$. All initial values in the sketch array and counter array are set to zero. The size of w represents the base number that will be used to update sensor data into the array. The size of d represents the number of digits of the number base w. That is, the number of possible values or sub-ranges that would represent values from sensor measurements is equal to w^d.

We also assume that the upper and lower bound of valid values of a measurement are pre-defined to each sensor. The measurement with values out of this valid range (R) will be simply judged as an outlier. The sensor will not transmit such readings and not update to the sketch array. Therefore, assuming an equal weight of all sub-ranges of the valid measurement, the resolution of sketched data (r) would be equal to the range of valid measurements divided by w^d, $r = R/w^d$. On the other hand, the resolution of sketched data is the error bound of the value estimation of a missing sample.

In the sketch process, we assume a window-based streaming process. That is, each sensor periodically updates a pre-defined number (N) of sensor readings, $(n_1, n_2, n_3, \ldots, n_N)$, into the fixed-sized sketch and counter array. This number is preset both at the base station and at sensor nodes, as it will be used as a period of cleaning process per sketch array. The number could be independent among different sensors and also dynamic, if needed. In this work, we only show a use of a fixed number of *N* in describing a simple concept of this work. Meanwhile, the use of the dynamic size of *N* needs a mechanism to optimize the value of *m* and coordinate it to both the base station and sensors.

We use a super-increasing set to update a sensor reading into a sketch array. A unique characteristic of the super-increasing set is that if there is a number, which is equal to a sum of members in a super-increasing set when no members are added more than one time, the set of members which produce the summation is unique [MH78]. Where the existing Count-Min sketch method [CM05] cannot recover a value of a specific missing sensor sample, we exploit this unique characteristic of the super-increasing set to be able to retrieve the missing sensor values matching with their corresponding temporal data such as their sampling sequence numbers.

Definition 1. *The super-increasing set (S) is the set that the value of members is a positive integer and value of the i^{th} member is greater than the sum of all the 1^{st} to $(i-1)^{th}$ members.*

$$S = \{s_i \mid [s_i > \sum_{k=1}^{i-1} s_k]\}$$

Lemma 1. *The minimum value of the i^{th} member in the super-increasing set (S) is equal to 2^{i-1}.*

$$S_{min} = \{s_i \epsilon\ S \mid s_i = 2^{i-1}\}$$

Proof: *The lemma 1 can be proven by using the induction proof. (1) The minimum value of the first member of set S is the minimum positive integer, $s_{1,min} = 1 = 2^{1-1}$. (2) The minimum value of the 2^{nd} member of set S is $s_{2,min} = 1+1 = 2^{2-1}$. (3) For $i \geq 3$, $s_i > s_{i-}$*

$_1$+…+s_2+s_1. *Because* $\sum_{m=0}^{M-1} 2^m = 2^M - 1$, *the minimum value of the* i^{th} *member of set S equal to* $s_{i,,min} = 2^{i-1}$

Before a sensor deployment phase, a super-increasing set with a size of m is also stored in both the base station and sensors where N must be divisible by m, which is the size of the super-increasing set $S=\{s_1, s_2, s_{3,\ldots,} s_m\}$. In the following analysis part, we will show that the maximum value of s_m is proportional to the energy spent in transmitting the sketch array. For the optimal energy cost, we will use S_{min} as our super-increasing set in the rest of this chapter.

To update a sensor reading into the sketch array, three operations need to be executed–(1) the operation to find a sub-range that would represent values of the sensor reading, (2) the operation to figure out a value to be added or updated into the sketch array and (3) the operation to figure out which components or members in the sketch array will be altered. In the first operation, a sub-range that represents values of the corresponding sensor reading (n_i) will be computed. As we assume that all sub-ranges in the valid range (R) are equally weighted, the suitable sub-range (r_{sub}) then simply equals $\lceil n_i/r \rceil$.

In the second operation, we use the sequence order of the sensor reading, not the value of the reading. For a sensor reading with a sequence order i, where $i \in [1,..,N]$, the value that would be updated into the sketch array (v_i) equals the super-increasing member of which its order in the set equals the sequence order of sensor readings (i) modulo with the size of the super-increasing set (m). That is, $v_i = s_{(i \bmod m)}$.

Thirdly, to find out what array members will be updated, we use the value of the suitable sub-range (r_{sub}) calculated in the first operation. The set of array members that will be updated is $U = \{a(p,q) \mid q = \lfloor r_{sub} / w^{p-1} \rfloor \, mod \, w\}$, where $p \in [1,..,d]$ and $q \in [1,..,w]$. Then, for a sensor reading (n_i), it will be updated into the sketch array, as shown in Equation 5.1.

$$\begin{aligned} a(p,q) &\leftarrow a(p,q) + v_i && when \;\; a(p,q) \in U \\ a(p,q) &\leftarrow a(p,q) && when \;\; a(p,q) \notin U \end{aligned} \tag{5.1}$$

Simultaneously, the corresponding counter array $c(p,q)$ will be updated accordingly, as shown in Equation 5.2.

$$\begin{aligned} c(p,q) &\leftarrow c(p,q) + 1 && when \;\; a(p,q) \in U \\ c(p,q) &\leftarrow c(p,q) && when \;\; a(p,q) \notin U \end{aligned} \tag{5.2}$$

Here, we demonstrate an example of how to update sensor data into a sketch array. To simplify, we will use w with size of 10 and d with size of 2, assuming that the cleaning system sets the range of valid measurements *(R)* at [0,200) and needs the sketch resolution within ±2 units. Also, assume that we are periodically cleaning a stream of a sensor every 20 samples and using a super-increasing set with size of $m=5$, {1, 2, 4, 8, 16}, for this example scenario. Then, the initial sketch and counter arrays would be the same and could be perceived as Figure 5.1.

.

Array	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Figure 5.1 Initial sketch and counter arrays with dimension 2x10

Array	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	*16*	*1*	*0*	*4*	*0*	*0*	*8*	*0*	*0*	*0*
1	*8*	*20*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Sketch array

Array	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	*1*	*1*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*0*
1	*1*	*2*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Counter array

Figure 5.2 The sketch and counter array updated by the 8th-11th sample

For example, if the 8th-11th sensor samples are 25, 12, 20 and 41, respectively, for the first operation, the suitable sub-range of each sensor sample is then 13, 6, 10 and 21, respectively. Secondly, the value that will be updated into the sketch array for each sensor sample is 4, 8, 16 and 1. Then, by the third operation, members in the sketch array will changed as $a_{0,3}$ and $a_{1,1}$ by the 8th sample, $a_{0,6}$ and $a_{1,0}$ by the 9th sample, $a_{0,0}$ and $a_{1,1}$ by the 10th sample and $a_{0,1}$ and $a_{1,2}$ by the 11th sample. Finally, the array members will be updated, as shown below. The sketch and counter array updated by the 8th-11th sensor sample can be illustrated, as in Figure 5.2.

$a_{0,0}$ =16, $a_{0,1}$ =1, $a_{0,3}$ =4, $a_{0,6}$ =8, $a_{1,0}$ =8, $a_{1,1}$ =4+16 =28, $a_{1,2}$ =1 and others remain 0. Also, $c_{0,0}$=1, $c_{0,1}$=1, $c_{0,3}$=1, $c_{0,6}$=1, $c_{1,0}$=1, $c_{1,1}$=2 and $c_{1,2}$=1 and others remain 0.

Cleaning process: For a particular sensor, the cleaning process is executed at the base station after the base station receives *N* sensor samples, as well as the corresponding sketch and counter arrays from the sensor. If there is no missing or corrupted sample detected, the cleaning process for these *N* samples is skipped.

The cleaning process is indeed a reverse calculation of the sketch process. For the available samples, which are received and not corrupted at the base station, the cleaning process will perform the second and third operations of the sketch process to figure out the value of v_i and the corresponding sketch member *a(p,q)*. Instead of adding v_i, the cleaning process deducts v_i and deducts by 1 out of the values in the corresponding *a(p,q)* and *c(p,q)*, respectively. Then, the remaining values in the sketch and counter arrays are those of missing samples. For each member of the sketch array that the remaining values is not equal to 0, the suitable sub-range value (r_{sub}) of each missing sample could be retrieved by algorithm in Table 5.1.

However, if there are any two or more missing samples that have been paired with the same v_i, the retrieval r_{sub} value could have multiple solutions and the cleaning process will not be able to recover the value of missing samples. Once the value of r_{sub} of each missing samples is retrieved, the value of the missing sample could then be computed by reverting the first operation of the sketch process.

For example, if the 8th and 10th sensor samples are missing but the 9th and 11th data are correctly received, the process will deduct the value of $v_9 = s_{(9 \bmod 5)}$ from $a_{0,6}$ $a_{1,0}$ and $v_{11} = s_{(11 \bmod 5)}$ will be deducted from $a_{0,1}$ and $a_{1,2}$.The remaining values are $a_{0,0}$ =16, $a_{0,3}$ =4 and $a_{1,1}$ =20. Because we know that there are two missing samples, it could be simply extracted that the value of $a_{1,1}$ is uniquely an addition of s_3 =4 and s_5=16. Then, we can recover the suitable sub-range values of the missing samples, which are 10 at sequence order of (5 mod 5) = (10 mod 5), and 13 at the sequence order of (13 mod 5) = (8 mod 5).

Lastly, by multiplying with the resolution which equals 2, the estimated missing value is 26 for the 8th sample and 20 for the 10th sample.

Note that the error of the estimated value could be at most ±2 units, which are the sketch resolution preset at the beginning of sensor deployment. In addition, the cleaning process cannot recover the values of any two or more missing samples, when the results of their sequence numbers modulo by *m* are the same. The details of this limitation will be discussed in the following section.

Table 5.1: Retrieve sub-range of a missing sample

```
Input:  (1) packet sequence of missing sample (miss_seq),
        (2) super-increasing set (S),
        (3) Sketch array (A),
        (4) Counter array (C)
Output: (1) sub-range of missing sample ($r_{sub}$)
IF miss_seq mod size(S) ≠ 0 THEN
   si_seq = miss_seq mod size(S)
  ELSE si_seq = size(S) END
FOR digit = 1 to d
  FOR num = 1 to w
      FOR j = size(S) to 1
          $b_j$ ← A(digit,num)/S(j)
          A(digit,num) ← A(digit,num) mod S(j)
      END
      WHILE $\sum_{j=1}^{m} b_j \neq C(digit, num)$ DO
```

```
            Set k equal to maximum index where $b_k$>0
            $b_k \leftarrow b_k$-1
            $b_{k-1} \leftarrow b_{k-1}$+2
        END
        append S(k) to Missing component set for $\forall k\{b_k \neq 0\}$
        IF S(si_seq) ϵ Missing component set THEN
            add (w^(digit-1))*(num-1) to $r_{sub}$
            skip to the next digit
        END
    END
END
```

5.4 Evaluation and Analysis

5.4.1 Analysis of Cleaning Performance

As mentioned, the cleaning process cannot recover when two or more missing sensor samples have been paired in the sketch process with the same member of the super-increasing set during sketch process due to the chance of obtaining the incorrect result of the extraction procedure. We use a metric, called cleaning performance, to define a capability that a method can correctly recover the value of missing data. The following is the analysis of the cleaning performance of the presented method.

Let *N* be the number of sensor samples that are sketched into a fixed array of memory, *m* be the size of the pre-defined super-increasing set, *p* be the probability that a sample will be missing, named as missing rate, and *X* be the number of missing samples that are matched with a particular member of the super-increasing set. To simply demonstrate the proof of concept, let *N* be divisible by m.

The probability that the missing samples are not mapped with the same member of the super-increasing set can be shown in Equation 5.3.

$$P\{X = 1\} = \binom{N}{m} p(1-p)^{(\frac{N}{m})-1} \tag{5.3}$$

Similarly, the probability that there are at least two missing samples which are paired with the same member of the super-increasing set can be computed in Equation 5.4.

$$P\{X \geq 2\} = \sum_{i=2}^{N/m} \binom{N/m}{i} p^i (1-p)^{(\frac{N}{m})-i} \tag{5.4}$$

Then, the cleaning performance (*C*) can be calculated as shown in Equation 5.5.

$$C = \frac{\binom{N}{m} p(1-p)^{(\frac{N}{m})-1}}{\binom{N}{m} p(1-p)^{(\frac{N}{m})-1} + \sum_{i=2}^{N/m} \binom{N/m}{i} p^i (1-p)^{(\frac{N}{m})-i}}$$

$$= \frac{\binom{N}{m} p(1-p)^{(\frac{N}{m})-1}}{\sum_{i=1}^{N/m} \binom{N/m}{i} p^i (1-p)^{(\frac{N}{m})-i}} \tag{5.5}$$

Since the cleaning performance is a function of the missing rate and the *N/m* ratio, Figure 5.3 shows the cleaning performance in the function of the missing rate with varying *N/m* ratios.

However, in fact, a sketch packet transmitted after each set of *N* sensor samples could also be missing. In this case, the missing samples in which the corresponding sketch packet is missing could not be recovered. The cleaning performance would be decreased by the factor of the missing rate, as shown in Equation 5.6.

$$C = p \, . \frac{\left(\frac{N}{m}\right) p(1-p)^{(\frac{N}{m})-1}}{\sum_{i=1}^{N/m} \binom{N/m}{i} p^{i} (1-p)^{(\frac{N}{m})-i}} \tag{5.6}$$

Since the transmission of a sketch packet could be perceived as a transmission of a packet that contains information of *N* sensor samples, it is insightful to consider the cleaning performance compared with a simple retransmission of *N* sensor samples. As the retransmission of a packet can recover the value of a missing sensor sample only when it is not missing or corrupted, we therefore compare its cleaning performance with that of the presented method, as shown in Figure 5.4. Although it shows that the performance of the retransmission is higher than those of variations of the presented method, the presented method consumes significantly less energy in the packet retransmission, as described in details in the following analysis section.

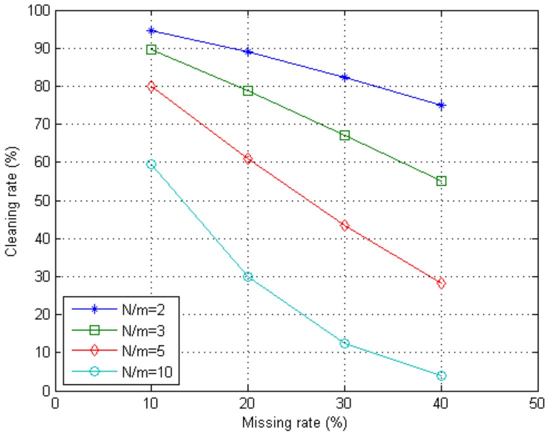


Figure 5.3 The cleaning performance in the function of the missing rate with varying *N/m* ratios.

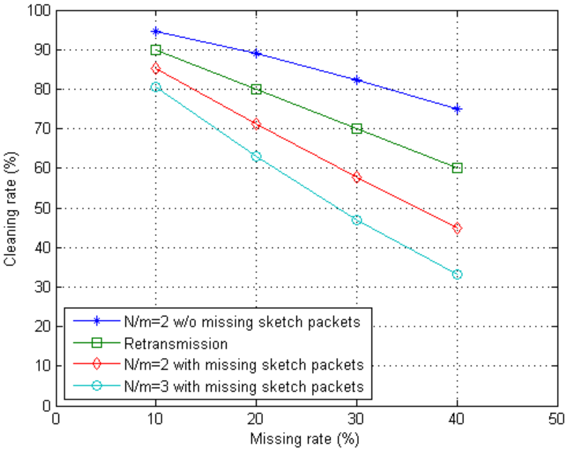


Figure 5.4 The cleaning performance of the sketch-based method and the simple retransmission.

5.4.2 Analysis of the Communication Cost

In this section, we analyze the communication cost in term of energy that the presented method spends for cleaning N sensor samples compared to that of the retransmission counterpart. In the analysis, we focus on the energy spent on a link between two neighboring sensors in transmitting and receiving packets for cleaning N sensor samples. We here adopt the energy model demonstrated in [SAM03]. The model of energy spent in transmission of an n-bit packet (E_n) is illustrated as follows.

$$E_n = (P_{te} + P_o)\left(\frac{\alpha+l+\tau}{R}\right) + (P_{tst}T_{tst}) + P_{re}\left(\frac{\alpha+l+\tau}{R}\right) + (P_{rst}T_{rst}) + E_{dec} \qquad (5.7)$$

where

$P_{te/re}$: *Power consumed in transmitter/receiver electronics*

$P_{tst/rst}$: *Start-up power consumed in the transmitter/receiver*

P_o : *Output transmit power*

R : *Transmission data rate (~20Kbps)*

E_{dec} : *The energy to decode the error correction per packet*

α *:The length of header bits per packet*

l *:The length of payload bits per packet*

τ *:The length of the trailer bits per packet*

To simplify the analysis, we assume the packet transmission without an error correction scheme which makes τ and E_{dec} equal zero. Then, the Equation 5.7 can be simplified in terms of radio parameters k_1 and k_2 as

$$E_n = k_1(\alpha + l) + k_2 \tag{5.8}$$

, where k_1 and k_2 are derived as $k_1 = \left(\frac{P_{te}+P_o+P_{re}}{R}\right)$ and $k_2 = (P_{tst}T_{tst} + P_{rst}T_{rst})$. Based on the RFM-TR1000 transceiver equipped in Mica sensors, $k_1 \approx 1.85$ *μJ/bit* and $k_2 \approx 24.86$ *μJ* [SAM03].

In the case of the presented method, only one sketch and one counter array would be transmitted for cleaning *N* sensor samples. For a counter array, the maximum number of each counter is at *N*. Thus, each member of the counter array needs to spare $\lceil log_2 N \rceil$ bits for counting the number of updates of each array member.

For a sketch array, size of the pre-defined super-increasing set is proportional to the size of a sketch array member. As shown in proof of Lemma 1, the minimum value of the addition of all members of the super-increasing set with size of *m* equals to 2^m-1. However, for cleaning *N* sensor samples, all members of a super-increasing set could be added up to $\left\lceil \frac{N}{m} \right\rceil$ times. Each array then needs to be able to contain values, which are at $\left\lceil \frac{N}{m} \right\rceil * (2^m - 1)$. Thus, the minimal size of an array member equals $\left\lceil log_2((2^m - 1) * \left\lceil \frac{N}{m} \right\rceil)\right\rceil = \left(m + \left\lceil log_2 \left\lceil \frac{N}{m} \right\rceil \right\rceil\right)$ bits.

Since a set of a sketch and counter array consists of $w * d$ members, the size of a set of sketch and counter arrays are then equal to $w * d * \left(m + \left\lceil log_2 \left\lceil \frac{N}{m} \right\rceil \right\rceil + \lceil log_2 N \rceil \right)$ bits. Thus, energy used in micro Joules for a link transmission of an *l*-bit packet, when α is assumed at 16 bits [SAM03], is given as follows.

$$E_n = 1.85\left(16 + w * d * (m + \left\lceil log_2 \left\lceil \frac{N}{m} \right\rceil \right\rceil)\right) + 24.86 \tag{5.9}$$

Meanwhile, for the retransmission case, the retransmission of *N* samples needs to consume energy, as shown in Equation 5.10, assuming that each packet contains an 8-bit sensor data and a 16-bit header.

$$E_n = N * [1.85(16 + 8) + 24.86] \tag{5.10}$$

Then, in cleaning *N* sensor samples, the energy consumption between the sketch-based method and the simple retransmission can be illustrated in Figure 5.5.

It is obvious that the retransmission method would consume less energy if the number of retransmission packets (N_{re}) is reduced to be less than the number of *N* sensor samples, $N_{re} < N$. However, to do so, the cleaning performance of the retransmission method would also decrease. Figure 5.6 demonstrates the cleaning performance between the sketch-based method with $N/m = 2$ and 3, and that of the retransmission method, which consumes energy equal to that of the sketch-based method.

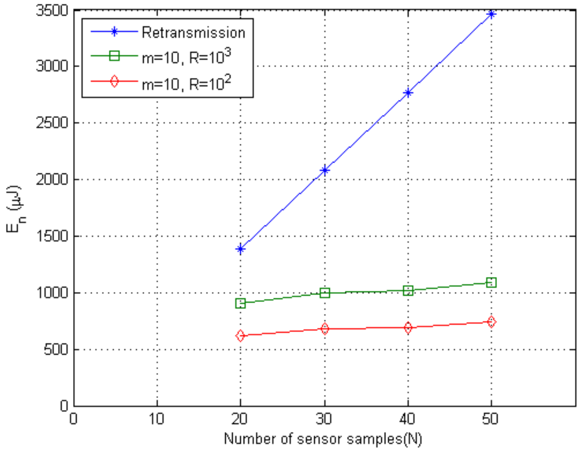


Figure 5.5 Comparison of energy consumption in cleaning *N* sensor samples among the sketch-based method with $R=10^2$, $R=10^3$ and the retransmission.

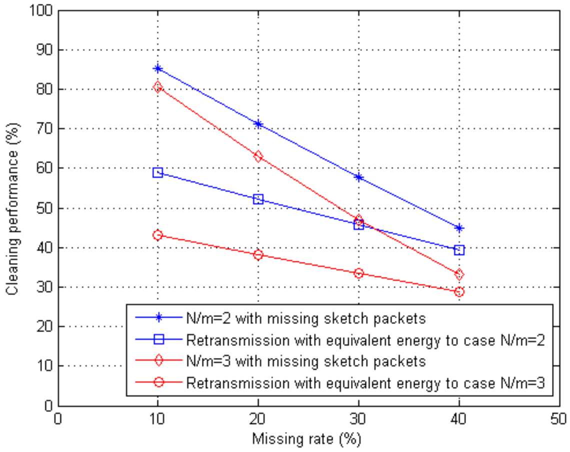


Figure 5.6 Cleaning performance of the sketch-based method compared with that of the retransmission method with adjusted energy consumption.

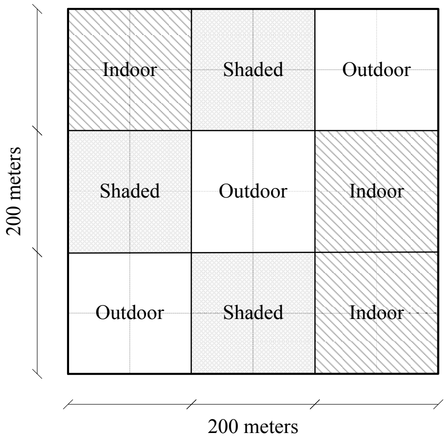


Figure 5.7 Layout of tested area of interest

5.4.3 Evaluation on Synthetic Data

In this section, we evaluate the cleaning performance of the presented method with a set of synthetic data of mobile sensors measuring temperature in a 200 x 200 square meter area. This area consists of 9 sub-areas with 3 different mean values of temperature– indoors, shaded and outdoors area – as shown in Figure 5.7.

The average temperature difference between the indoors and shaded area is about 6 degree Celsius, and that between shaded and outdoors area is about 7 degree Celsius. The temperature in each area is evolving according to the data observed at Asheville regional airport during Jan 1-15, 2007 [Nat02].

We experimented on mobile sensors, which move according to various mobility models–random waypoint, nomadic and random street [CBD02,Bon02]. The traces of sensor movements are generated by using the Bonnmotion mobility generator [Bon02] with different node speeds – 10, 20 and 30 meters/minute.

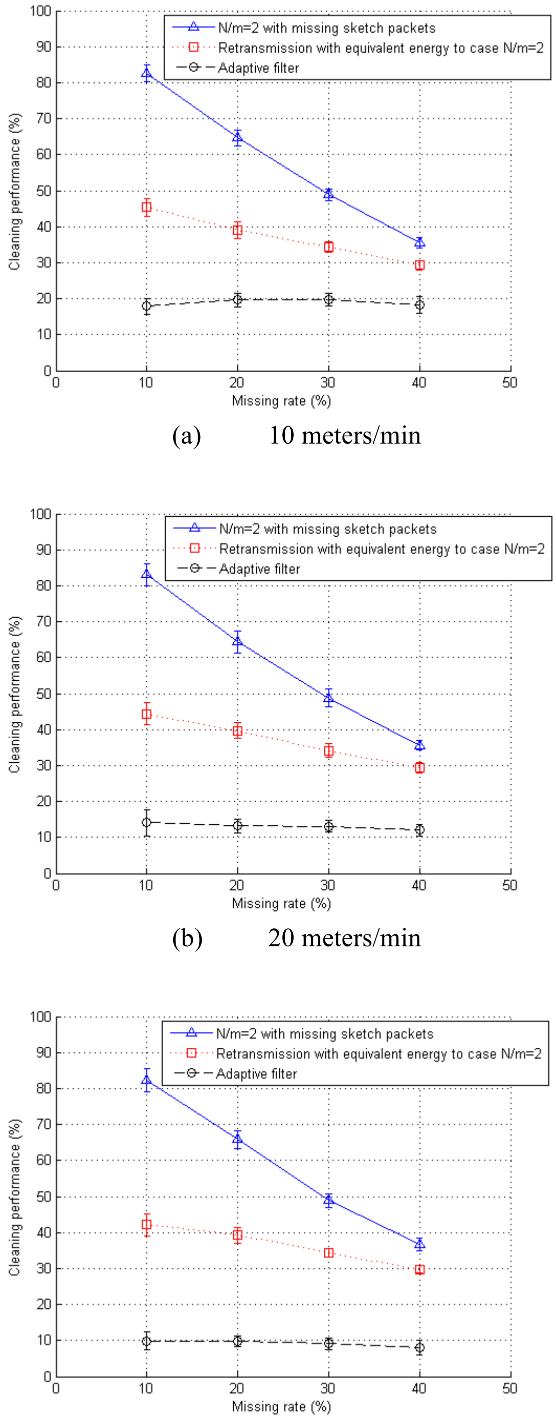
As we assumed that data from neighboring sensors are not valid or available to use for cleaning a sensor sample, we then compared the cleaning performance of the sketch-based method with those of the existing data cleaning methods that use only temporal data of a sensor, not data from other sensors. Besides the retransmission counterpart with the same energy consumption, we also compared the proposed method with the cleaning method using adaptive filter [5], which is one of the most popular methods to predict the value of missing samples by using only temporal data of a specific sensor. In addition, we did not consider the spatio-temporal based techniques because their mechanisms to select the associated neighboring sensors for the cleaning process are not practical to the transient environment in mWSNs. The comparison result is shown in Figure 5.8.

From Figure 5.8, we found that the cleaning performance of the sketch-based method is not affected by the speed and mobility pattern of mobile sensors. The overall performance is lower than that of the theoretical performance as expected due to the probability of the incorrect component extraction in the cleaning process. However, the performance remains higher than those of adaptive filter and the retransmission methods when they consume the same energy to retransmit packets for cleaning.

In fact, the adaptive filter based cleaning method does not require sensors to transmit any additional packets for the purpose of cleaning. However, the results demonstrate that it performs poorly and worse when sensor nodes move faster. That is, the prediction cannot cope with the reading fluctuations especially when sensors move across the sub-areas. Meanwhile, the sketch-based method needs additional energy consumption in sensor nodes but it is not affected by changes of sensor speeds or fluctuations in readings.

5.5 Summary

In this chapter, we have presented a novel sketch-based data cleaning method to recover the values of missing samples in a sensor data stream. The sketch-based cleaning method relies only on a sketch packet, which plays a role in a summary of *N* sensor samples. It does not rely on data from other nearby sensors with any types of contextual relationships. It requires a small portion of additional power consumption to transmit a sketch packet compared to that of transmission of the original sensor data for the increase of ability to recover the missing or corrupted sensor data. Meanwhile, this work can be applied to the static WSNs. We focused our design to clean data for mWSNs where sensor data from different sensors do not demonstrate any forms of data correlation.



(a) 10 meters/min

(b) 20 meters/min

(c) 30 meters/min

Figure 5.8 Cleaning performance of the sketch-based method compared with those of the adjusted-energy consumption retransmission method and the temporal adaptive filter with various average sensor speeds

CHAPTER 6

DISCUSSION, FUTURE WORKS AND CONCLUSION

This dissertation described methods to clean the real-time sensor data streams in different system restrictions. These include data cleaning methods in dense and sparse mWSNs, as well as a method to clean data when sensors are not observing a shared environment. In this chapter, we discuss the contributions and limitations of what we achieved, outline future works and conclude this dissertation.

6.1 Virtual Sensor for Mobile Sensor Data Cleaning

In Chapter 3, we have presented a novel method to clean missing data in mWSN applications. This method applies the concept of virtual sensor combining with an adaptive filter. When the location information of the sensor is precisely and completely received at the base station, this method can cleans dirty samples more than 80% of missing data. The average speed of sensor nodes does not affect the cleaning performance. In addition, the method does not need any additional hardware implementations. This method can be efficiently implemented in scenarios, where the area of interest is temporary for operations.

Furthermore, a requirement to gain an efficient cleaning performance is that the coverage area and location of a VS must be well defined. In particular, each coverage area should be an area where the desired measurements are expected to be similar, i.e., the variance of measurements is not expected to be large. The more precisely the system administrators can identify the boundary of such dissimilarity of data features, the higher the cleaning performance they can achieve.

Although the VS method successfully cleans mobile sensor data, it is limited in that a priori knowledge of the coordinates that define the VS boundaries is needed to attain a satisfactory cleaning performance. Since site surveys can be fuzzy or not well developed for locations without a permanent presence, boundary coordinates are often naturally indistinct, or too costly. This is a considerable limitation of this work.

6.2 Belief-based Cleaning in Trajectory Sensor Streams

In Chapter 4, we have presented another novel, but simple method of data cleaning suited to mWSN applications. Rather than relying on the static spatio-temporal relationships among sensors, which is invalid to mWSNs, we analyzed sensor trajectories relative to pre-defines area as the residence pattern and existence frequency to indicate how a neighboring sensor can help in the cleaning process. In addition, we also introduced the cumulative detection rate of dirty data to quantify the reliability level of a data stream within a particular sub-area. We evaluated the cleaning performance under scenarios with various mobility models, dirty data rate and average sensor speed. The cleaning performance of this method is superior to those of the existing cleaning methods.

Although the performance of the belief-based method outperforms those of its peers, there are still constraints to overcome. First, the interpretation of the desired context still mainly relies on a priori-knowledge of sub-area boundaries. This is the same constraint as that in the virtual sensor-based method, as stated above. Second, the trajectory readings must be precise and complete. In reality, location reports always contain an extent of uncertainty. Furthermore, another major constraint is that the design and evaluation still consider sensors that monitor the shared environment. If the sensors, for example, are measuring the stress level of soldiers, this method will not provide a good cleaning performance.

6.3 Sketch-based Cleaning in Sensor Data Streams

In Chapter 5, we have presented a novel sketch-based data cleaning method to recover the values of missing samples in a sensor data stream. This method utilizes a unique feature of the super-increasing set to summarize information of *N* sensor samples into a sketch packet. The sketch packet then plays an important role as a summary of *N* preceding sensor samples and it could be used to recover corrupted or missing samples at the base station.

This method does not rely on data from other nearby sensors with any types of contextual relationships. Therefore, it is suitable to scenarios, where sensors are distant from each other, including where sensors are not measuring a shared environmental phenomenon. Although this method was designed for mWSNs, it can be applied to static WSNs since the cleaning mechanism of a sensor stream utilizes information only from the corresponding sensor stream.

Since this method can clean sensor data when the corrupted data samples are not sharing the same sequence of sketching process, the cleaning performance will rapidly decrease when all types of interference causes a burst of errors to transmitted data. Furthermore, this method requires an additional power consumption to transmit the sketch packets for recovering the missing or corrupted sensor data, compared to existing methods that rely on spatio-temporal correlations of sensors. It is then more proper to clean sensor data in mWSNs when other types of contextual relationships among sensors cannot be resolved.

6.4 Future Work

In spite of what we achieved, there are more challenges in the field of data cleaning in mWSNs. We here outline the potential research directions that can enhance our current works for more comprehensive data cleaning mechanisms in mWSNs as follows.

6.4.1 Design and develop a semantic trajectory relationship model

One research direction is to build a trajectory-based relationship model that illustrates how the physical, spatial-temporal, symbolic, absolute, relative contexts of trajectories affect correlations among sensor data in mWSNs. A semantic trajectory relationship model could be designed to support the development of a sensor selection process by providing the definition and format of dependencies among trajectory relationships. To develop this model, it first needs to extract semantically meaningful trajectory relationships from spatio-temporal instances. Then, one can investigate how each semantic trajectory relationship can be used to determine the structural and behavioral correlations among sensor data streams.

To analyze the trajectory of a mobile sensor, existing solutions represent an object's trajectory as an ordered list of location samples at specific instances in time [GH05]. Although such a list can adequately represent the changing positions of an object throughout its lifespan, it does not contain data necessary to segment an object's trajectory into semantically meaningful trajectory relationships, such as moving together, following and overtaking one another. Even if the ordered list of location samples was mechanically broken down into smaller sets of ordered lists of locations, it would still not provide any insight into the behavioral or structural meaning of this list unless semantic relationships among trajectories were identified.

To identify relationships among trajectories of different sensors, one needs first extract the intended semantics from individual sensors' trajectories. Second, an analysis of these semantics to identify meaningful relationships among trajectories of different sensors is needed. The promising design could integrate spatio-temporal reasoning operators [All83,GN02] with structural and behavioral semantics for annotating trajectories. Although a simple framework for semantic trajectory annotation has been developed [TCP$^+$11], this framework lacks a mechanism that identifies trajectory relationships between multiple sensors' trajectory semantics. Yet, a comprehensive mechanism that does so is critical for developing a sensor selection process for data cleaning.

As mentioned, the sensor selection process is a primary issue that is not well addressed in existing sensor data cleaning methods. The existing methods currently rely only on an associated set of static sensors to help in cleaning processes. However, when sensors are moving in mWSNs, we can no longer rely on such a pre-defined static set of helpful sensors. These methods are thus not applicable to mWSNs. To solve this issue, one can perceive a data cleaning method that is grounded in the dynamic correlations between sensor data streams and semantic trajectory relations. The hypothesis of this research direction is that if the relationships between semantic sensor trajectories are identified, they can be used to determine the correlations between sensor data streams of moving sensors that are sensing a phenomenon related to a shared environment. If identified, a semantic trajectory relationship model could be designed to support the development of a sensor selection process by providing the definition and format of dependencies among trajectory relationships. Therefore, determining structural and behavioral dependencies among trajectories will result in a model that denotes streams, not just as an ordered sequence of points, but also as a sequence of points annotated with semantics.

6.4.2 Design and develop a dynamic trajectory and context-aware annotation method for trajectory sensor streams.

While semantic trajectory relationships provide insight into the structural and behavioral aspects of sensors, they do not reflect the current state of the environment or the entity that houses the sensor. To illustrate the importance of solving this issue, consider the following example: suppose sensor A and B share a semantic trajectory relationship (e.g., sensor A and sensor B follow the same trajectory and arrive at a given destination at the same time), it does not by default mean that these two sensors operate under the same sensing environment (e.g., sensor A senses humidity and sensor B senses temperature). Although sensor A and B share a semantic trajectory relation, they might not operate under the same context. More specifically, if each sensor monitors the body temperature of different officers, a difference in measurement could be due to the fact that one officer was jogging while the other had a fever.

The objective of this direction is to design and develop a method that dynamically gives context to semantic trajectory relationships and sensing environments. Therefore, determining structural and behavioral dependencies among trajectories will result in a model that denotes streams, not just as an ordered sequence of points, but also as a sequence of points annotated with semantics. In contrast to the previous direction, in this stage one should focus on cleaning sensor data using sensors' surrounding environments and contexts. The context should not be based on the spatio-temporal or other sematic features of sensor trajectories, although one could later integrate sematic trajectory relationships and context to form a comprehensive sensor selection mechanism for data cleaning.

One way to accomplish this objective is that one could develop a method to select the contexts of sensor operations, which can be used to extract knowledge about the sensor streams of interest. Then, one could continue by determining how to transform the raw context information into meaningful semantics. Later, one could develop a method to efficiently annotate semantics with trajectory-sensor streams. The classification concept of trajectory relations might be helpful and deployed for cleaning sensor data in scenarios in which sensors are measuring a shared spatio-temporal phenomenon. Moreover, one should examine context awareness separately from semantic trajectory relationships because there are scenarios in which sensor streams' correlations can only be identified based on context alone rather than trajectory semantics. For example, there are sensing environments in which sensor reports would be better correlated based on context, such as soldier activity, exposure to blasts, and transportation method, as opposed to trajectory relationships. For example, the readings of two soldiers' health sensors are not necessarily reflected by the proximity of the soldiers. Instead, the readings might be reflected by other "context semantics," such as the number of hours a solider sleeps per night or stress level of a soldier.

A hypothesis of this research direction is that there are application domains in which context awareness or semantic trajectory relationships are sufficient for data cleaning, and there are other domains in which it is necessary to consider both context awareness and semantic trajectory relationships. If this hypothesis proves to be true, the end result of this phase of this research direction will be a comprehensive sensor selection tool for data cleaning that considers three scenarios noted above. Otherwise, a novel sensor selection process for each of the scenarios listed above needs to be separately developed.

6.4.3 Design and develop a comprehensive data cleaning method that tolerates the uncertainty of trajectory readings.

The main approach of this direction is to develop a comprehensive data cleaning method for mWSNs that integrates context awareness and semantic trajectory relationships. On the one hand, one can assume that mobile sensing devices report their precise location and move directly into the development of the solution. On the other hand, in practice, noisy sensor readings generate imprecise localization data, which in turn reduces the accuracy of trajectory data. A data cleaning method should either tolerate the uncertainty of trajectory readings or address this issue prior to implementing a comprehensive data cleaning solution.

Besides the inborn locality imprecision of sensing device, many phenomena, such as the limited power of radio transceivers, battery outages, and indoor signal losses, can lead to GPS data loss. The uncertainty of the trajectory data prevents an accurate classification of the trajectory relations and the interpretation of helpful semantics. Therefore, in this research direction, one should focus on developing an efficient online trajectory extraction technique for estimating imprecise sensor locations that tolerates trajectory uncertainty. It is expected that an insight from this research into semantic trajectory relationships and context awareness will help identify which groups of trajectories are most similar and therefore provide a basis for assessing the degree of uncertainty of individual trajectory readings. Once the uncertainty has been assessed, one could explore the feasibility of incorporating machine learning based sensor cooperation solutions to provide estimation schemes that reinstate the values of corrupted sensor data with uncertain trajectory data. The accomplishment of this future work will deliver the data cleaning solution that can tolerate the uncertainty of trajectory data reported by mobile sensors that are measuring phenomena of both the shared environment and non-shared environment.

6.5 Conclusion

Modern mobile applications are emerging and exploiting all kinds of sensors. These applications will become more seamless to our daily lives and enhance life quality of humankind. However, these applications will be not viable unless the imprecision of the collected sensor data is corrected. This dissertation stems from the recognition of the significant need in data cleaning in mWSNs. Data cleaning methods for mWSNs scenarios that consider dynamic sensor characteristics and relationships are investigated and developed. There remain more challenges in this area. We hope our fulfilling work will benefit the emerging mobile sensor applications and inspire scholars to accomplish more comprehensive data cleaning solutions in mWSNs.

REFERENCES

[ABB^{+}03] Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J. and Widom, J. (2003). Stream: The Stanford Stream Data Manager (demonstration description). *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. 656-665.

[ABN08] Abul, O., Bonchi, F., Nanni, M. (2008). Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. *ICDE*. 376-385.

[AFM06] Andritsos, P., Fuxman, A., & Miller, R. (2006). Clean answers over dirty databases: a probabilistic approach. *Proceedings of the 22nd International Conference on Data Engineering.* doi: 10.1109/ICDE.2006.35

[Agg02] Aggarwal., C. (2002). An intuitive framework for understanding changes in evolving data streams. *Proceedings of IEEE ICDE conference*.

[All83] Allen, J. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*. 832–843.

[ALMS04] Abadi, D., Lindner, W., Madden, S., & Schuler, J. (2004). An integration framework for sensor networks and data stream management systems. *Proceedings of the 30th International Conference on Very Large Data Bases Endowment*, 30. 1361-1364.

[AS10] Aschenbruck, N., & Schwamborn, M. (2010). Synthetic map-based mobility traces for the performance evaluation in opportunistic networks. *Proceedings of the 2nd ACM International Workshop on Mobile Opportunistic Networking*. Pisa, Italy.

[AWSC02] Akyildiz, I. F., Weilian, S., Sankarasubramaniam, Y., & Cayirci, E. (2002).A survey on sensor networks. *IEEE CommunicationMagazine, 40*(8). 102-114.

[BCFL09] Bu, Y., Chen, L., Fu, W., & Lui, D. (2009). Efficient anomaly monitoring over moving object trajectory streams. *Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.159-168.

[Bon02] BonnMotion – a mobility scenario generation and analysis tool. University of Bonn, Germany. [Online]. Available: http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/. (2002)

[BPM08a] Bakhtiar, A., Pissinou, N., & Makki, K. (2008). Approximate replication of data using adaptive filter in wireless sensor networks. *Proceedings of* *the 3rd International Symposium on Wireless Pervasive Computing (ISWPC 2008)*. 365-369.

[BPM08b] Bakhtiar, A., Pissinou, N., & Makki, K. (2008). Estimated replication of data in wireless sensor networks. *Proceedings of the 3rd International Conference on Communication System Software and Middleware*. 107-110.

[BPM09] Bakhtiar, Q. A., Pissinou, N., & Makki, K. (2009). Belief based data cleaning for wireless sensor network. *Wireless Communications and Mobile Computing.* doi: 10.1002/wcm.970

[CBD02] Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2, 483-502.

[CCX08] Cheng, R., Chen, J., & Xie, X. (2008).Cleaning uncertain data with quality guarantees. *Proceedings of ACM Very Large Data Bases Endowment. 1*(1). 722-735.

[CDTW00] Chen, J., Dewitt, D., Tian, F., & Wang, Y. (2000). NiagaraCQ: A scalable continuous query system for internet databases. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 379-390.

[CG08] Cormode, G. & Garofalakis, M. (2008). Approximate continuous querying over distributed streams. *ACM Transactions on Database System (TODS)*.

[CFG+07] Cong, G., Fan, W., Geerts, F., Jia, X., & Ma, S. (2007). Improving data quality: consistency and accuracy. *Proceedings of the 33rd International Conference on Very Large Data Bases Endowment*. 315-326.

[CM05] Cormode, G. and Muthukrishnan, S. (2005). An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*. 58-75.

[CS89] Cochran, W. & Snedecor, G. (1989). Statistical Methods. Iowa State University Press.

[DGMH04] Deshpande, A., Guestrin, J. H. C., Madden, S., & Hong, W. (2004). Model-driven data acquisition in sensor networks. *Proceedings of the 30th International Conference on Very Large Data Bases.* 588-599.

[DH00] Domingos, P., Hulten., G. (2000). Mining high-speed data streams. *Proceedings of ACM SIGKDD*.

[EGH+08] J. Eriksson et al., "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in Proc. of the 6th ACM International Conference on Mobile Systems, Applications and Services, Breckenridge, CO, USA, June 2008.

[EN03] Elnahrawy, E., & Nath, B. (2003). Cleaning and querying noisy sensors. *Proceedings of the 2nd ACM International Workshop in Wireless Sensor Network.* 78-87.

[EN04] Elnahrawy, E., & Nath, B. (2004). Context-aware sensors. *Proceedings of European conference of Wireless Sensor Networks*. 77-93.

[FGT07] Frentzos, E., Gratsias, K., & Theodoridis, Y. (2007).Indexed-based most similar trajectory search. *Proceedings of the 23rd International Conference on Data Engineering.*816-825.

[GFS+01] Galhardas, H., Florescu, D., Shasha, D., Simon, E., & Saita, C. A. D. (2001). Declarative data cleaning: language, model and algorithms. *Proceedings of the 27th International Conference on Very Large Data Bases.* 371-380.

[GHS07] Gonzalez, H., Han, J., & Shen, X. (2007). Cost-conscious cleaning of massive RFID data sets. *Proceedings of the 23rd International Conference on Data Engineering.*1268-1272.

[GL06] Ganguly, S., and Lakshminath, B. (2006). Estimating entropy over data streams. In *Proceeding of the European Symposium on Algorithm (ESA)*.

[GN02] Gerevini, A., & Nebel, B. (2002). Qualitative spatio-temporal reasoning with RCC-8 and Allen's interval calculus: computational complexity. *Proceedings of the 15th European conference on artificial conference (ECAI).*

[GNPP07] Giannotti, F., Nanni, M., Pedreschi, D. & Pinelli, F. (2007).Trajectory pattern mining. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 330-339.

[GS05] Güting, R. H., & Schneider, M. (2005). Moving objects databases. Morgan Kaufmann.

[GTW+10] Ghica, O., et al. (2010). Trajectory data reduction in wireless sensor network. *International Journal of Next-generation Computing,* vol. 1, No.1.

[Hay02] Haykin, S. (2002). Adaptive Filter Theory, *Prentice Hall,* Englewood cliffs, NJ, 4th ed.

[Hay03] Haykin, S. (2003). Least mean square adaptive filters. *New York Wiley InterScience*.

[HBZ$^+$06] B. Hull, et al., "CarTel: a distributed mobile sensor computing system," in Proc. of the 4th International Conference on Embedded Networked Sensor Systems, Boulder, Colorado, USA, November 1-3, 2006.

[HGJT07] Huang, L., Garofalakis, M., Joseph A. D., Taft, N. (2007). Communication-Efficient Tracking of Distributed Cumulative Triggers, *Proceedings of the 27th International Conference on Distributed Computing Systems*.

[IKM00] Indyk, P., Koudas, N., Muthukrishnan, S. (2000). Indentifying Representative Trends in Massive Time Series Data Sets Using Sketches. In *VLDB Conference*.

[Ind03] Indyk, P. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. (2003). In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*.

[Int04] Intel Lab Data. (2004). http://db.csail.mit.edu/labdata/labdata.html

[JAF$^+$06] Jeffery, S., Alonso, G., Franklin, M., Hong, W., & Widom, J. (2006). A pipelined framework for online cleaning of sensor data streams. *Proceedings of the 22nd International Conference on Data Engineering.*

[JGF06] Jeffery, S. R., Garofalakis, M., and Franklin, M. J. (2006). Adaptive Cleaning for RFID Data Streams. *Proceedings of the 32nd International Conference on Very Large Data Bases*. 163-174.

[JHI07] Jayasumana, A., Han, Q. & Illangasekare, T. (2007). Virtual Sensor Networks – A Resource Efficient Approach for Concurrent Applications. *ITNG'07: International conference on information technology*.

[JL84] Johnson, W., Lindenstrauss, J. (1984). Extensions of Lipshitz mapping into Hilbert space. In *Contemporary Mathematics*. Vol. 26. 189-206.

[JLOZ06] Jensen, C., Lin, D., Ooi, B., & Zhang, R. (2006). Effective density queries on continuously moving objects. *Proceedings of the 22nd International Conference on Data Engineering*.

[JM96] Johnson, D., & Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile Computing, T. Imelinsky and H. Korth (Eds.)*, Kluwer Academic Publishers, Norwell, MA, 1996, 153-181.

[KBS06] Khoussainova, N., Balazinska, M., & Suviu, D. (2006). Towards correcting input data errors probabilistically using integrity constraints.

Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access. 43-50.

[KCR06] Keralapura, R., Cormode, G., Ramamirtham, J. (2006). Communication-efficient distributed monitoring of threshold counts, *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*.

[KPJ06] Kabadayi, S., Pridgen, A. & Julien, C. (2006). Virtual Sensors: Abstracting Data from Physical Sensors. *Proceedings of WoWMoM'06*.

[KSR08] Kumar, R., Shin, J. & Ramachandran, U. (2008). Mobile Virtual Sensors: A Scalable Programming and Execution Framework for Smart Surveillance. *ACM HotEmNets'08*.

[LHKG07] Li, X., Han, J., Kim, S., & Gonzalez, H. (2007). ROAM: Rule-and motif-based anomaly detection in massive moving object data sets. *Proceedings of the 7th SIAM International conference on Data Mining*.

[LHL08] Lee, J., Han, J., & Li, X. (2008). Trajectory outlier detection: a partition-and-detect framework. *Proceedings of the 24th International Conference on Data Engineering.*140-149.

[LHLG08] Lee, J., Han, J. Li, X., & Gonzalez, H. (2008). TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the 34th International Conference on Very Large Data Bases Endowment, 1*(1). 1081-1094.

[LHW07] Lee, J., Han, J., & Whang, K. (2007). Trajectory clustering: a partition-and-group framework. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data.*593-604.

[LPT99] Lui, L., Pu, C., & Tang, W. (1999). Continual queries for internet-scale event-driven information delivery. *IEEE transaction of knowledge and data engineering*, vol.11, 610-628.

[MH78] Merkle, R. C, and Hellman, M. E. (1978). Hiding information and signatures in trapdoor knapsacks. In *IEEE Transactions on Information Theory*. Vol. IT-24, No. 5.

[MNP10] Mayfield, C., Neville, J., & Prabhakar, S. (2010). ERACER: a database approach for statistical inference and data cleaning. *Proceeding of the 2010 International Conference on Management of Data.* 75-86.

[MPTG09] Monreale, A., Pinelli, F., Trasarti, R., & Giannotti, F. (2009). WhereNext: a location predictor on trajectory pattern mining. *Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 637-646.

[MSHR02] Madden, S., Shah, M., Hellerstein, M., & Raman, V. (2002). Continuously adaptive continuous queries over streams. *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*. 49-60.

[MWA+03] Motwani, R. et al. (2003). Query processing approximation and resource management in a data stream management system. *Proceedings of the 1st Biennial Conference on Innovative Data Systems Research*.

[MZM+09] Mo, Z., Zhu, H., Makki, K., Pissinou, N., & Karimi, M. (2009). On Peer-to-peer Location Management in Vehicular Ad Hoc Networks. *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, 1(2).

[Nat07] National Climate Data Center. (2007). Online Climate Data Directory, US Department of Commerce. Retrieved on January, 2010. http://cdo.ncdc.noaa.gov/qclcd/qclcdhrlyobs.htm

[NYZ+12] Nikzad, N., Yang, J., Zappi, P., Rosing, Tajana and Krishnaswamy, D. (2012). Model-driven Adaptive Wireless Sensing for Environmental Healthcare Feedback Systems. To be appear in *Proceeding of IEEE International Conference in Communications*.

[PP10] Pumpichet, S., & Pissinou, N. (2010).Virtual sensor for mobile sensor data cleaning. *Proceedings of the IEEE International Conference on Global Telecommunications*, 1-5.

[PPJP12] Pumpichet, S., Pissinou, N., X. Jin, & Pan, D. (2012). Belief based cleaning in trajectory sensor streams. To be appeared in *Proceedings of the IEEE International Conference on Communications.*

[PS07] Petrosino, A., & Staiano, A. (2007). A neuro-fuzzy approach for sensor network data cleaning. *Proceedings of the 11th International Conference on Knowledge-based and Intelligent Information and Engineering Systems*. 4697. 140-147.

[RDTC06] Rao, J., Doraiswamy, S., Thakkar, H., & Colby, L. S. (2006).A deferred cleansing method for RFID data analytic. *Proceedings of the 32nd International Conference on Very Large Data Base.* 175-186.

[Rew03] Rewienski, M. (2003). A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems. Doctoral thesis. Massachusetts Institute of Technology.

[SAM03] Sankarasubramaniam, Y., Akyildiz, I. F., and McLaughlin S. W. (2003). Energy efficiency based packet size optimization in wireless sensor networks. In *Proceeding of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*. 1-8.

[SBC+06] Sarlos, T., Benzur, A., Csalogany, K., Fogaras, D., and Racz, B. (2006). To randomize or not to randomize: space optimal summaries for hyperlink analysis. In *Proceedings of the International Conference on World Wide Web (WWW)*.

[SH05] Schilling, R., & Harris, S. (2005). Fundamentals of digital signal processing using MATLAB. Bill Stenquist, USA. 628, 652-653.

[SJFW06] Sarma, A. D., Jeffery, S. R., Franklin, M. J., and Widom, J. (2006). Estimating Data Stream Quality for Object-Detection Applications. *Proceedings of the 3rd International ACM SIGMOD Workshop on Information Quality in Information System*.

[SPP+06] Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V. & Gunopulos, D. (2006). Online outlier detection in sensor data using non-parametric models. *Proceedings of the 32nd International Conference on Very Large Data Bases*.187-198.

[SQW+09] Song, B., Qin, P., Wang, H., Xuan, W., & Yu, G. (2009). bSpace: a data cleaning approach for RFID data streams based on virtual spatial granularity. *Proceedings of the 9th IEEE International Conference on Hybrid Intelligent Systems.* 252-256.

[SR06] Santini, S., & Romer, K. (2006). An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks. *Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS '06).*

[STA+12] Sakurai, Y. et al. (2012). Towards sensor based context aware systems. *Sensors*. vol. 12. 632-649.

[TM06] Tulone, D., & Madden, S. (2006). PAQ: time series forecasting for approximate query answering in sensor networks. *The 3rd European Workshop on Wireless Sensor Networks, Switzerland.* 3868. 21-37.

[TPL+10] Tran, T., Peng, L., Li, B., Diao, Y., & Liu, A. (2010). PODS: a new model and processing algorithms for uncertain data streams. *Proceedings of the 2010 International Conference on Management of Data.*159-170.

[Tra11] Trajcevski, G. (2011). Uncertainty in Spatial Trajectories. *Computing with Spatial Trajectories*. 63-107.

[TTD$^+$09] Trajcevski, G., Tamassia, R., Ding, H., Scheuermann, P., Cruz, I. (2009). Continuous probabilistic nearest-neighbor queries for uncertain trajectories. *EDBT*. 874-885.

[VVS$^+$00] Vassiliadis, P., Vagena, Z., Skiadopoulos, S., Karaannidis, N., & Sellis, T. (2000). Arktos: a tool for data cleaning and transformation in data warehouse environments. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.* 28. 42-47.

[Yan09] Yan, Z. (2009). Towards Semantic Trajectory Data Analysis: A Conceptual and Computational Approach. *Proceedings of the Very Large Data Base PhD Workshop.*

[YCP$^+$11] Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., & Aberer, K. (2011), SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories. *Proceedings of the 14th ACM International Conference on Extending Database Technology (EDBT 2011)*. 259-270.

[ZCWL07] Zhuang, Y., Chen, L., Wang, X., & Lian, J. (2007). A weighted moving average-based approach for cleaning sensor data. *Proceeding of the 27th International Conference on Distributed Computing Systems.* doi: 10.1109/ICDCS.2007.83

[ZS02] Zhu, Y., and Shasha, D. (2002). StatStream: statistical monitoring of thousands of data streams in real time. *Proceedings of the 28th International Conference on Very Large Data Bases.*358-369.

VITA

SITTHAPON PUMPICHET

	Born, Bangkok, Thailand
2002	B.E., Electrical Engineering Chulalongkorn University Bangkok, Thailand
2002-2005	Network engineer, TOT Corporations Limited, Thailand
2007	M.S., Electrical Engineering San Jose State University San Jose, California
2013	Doctoral Candidate Florida International University Miami, Florida

PUBLICATIONS AND PRESENTATIONS

S. Pumpichet, X. Jin, and N. Pissinou, "Sketch-based Data Recovery in Sensor Data Streams," in *Proceedings of IEEE International Conference on Networks (ICON 2013)*, Singapore, December 11-13, 2013. (to appear)

X. Jin, N. Pissinou, S. Pumpichet, C. Kamhoua, and K. Kwiat, "Modeling Cooperative Selfish and Malicious Behaviors for Trajectory Privacy Preservation using Bayesian Game Theory," in *Proceedings of IEEE International Conference on Local Computer Networks (LCN 2013)*, Sydney, Australia, October 21-24, 2013. (to appear)

S. Pumpichet, N. Pissinou, X. Jin and D. Pan, "Belief-based Cleaning in Trajectory Sensor Streams," in *Proceedings of IEEE International Conference on Communications (ICC 2012)*, Ottawa, Canada, June 10-15, 2012, pp. 208-212.

X. Jin, N. Pissinou, C. Chesneau, S. Pumpichet, and D. Pan, "Hiding Trajectory on the fly," in *Proceedings of IEEE International Conference on Communications (ICC 2012)*, Ottawa, Canada, June 10-15, 2012, pp. 403-407.

S. Pumpichet and N. Pissinou, "Virtual Sensor for Mobile Sensor Data Cleaning," in *Proceedings of IEEE International Conference on Global Communications (GLOBECOM 2010),* Miami, USA, December 6-10, 2010, pp. 1-5.

N. Mir, S. Pumpichet and H. Chan, "An Efficient Inter-Domain Routing in Wireless Mesh Network," in *Proceedings of the 9th International Conference on Parallel and Distributed Computing and Networks (PDCN 2010)*, Paper Reference: 676-012, Innsbruck, Austria, February 16-18, 2010.