7-17-2020

Solving Complex Data-Streaming Problems by Applying Economic-Based Principles to Mobile and Wireless Resource Constraint Networks

Concepcion Z. Sanchez Aleman
Florida International University, csanc066@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY
Miami, Florida

SOLVING COMPLEX DATA-STREAMING PROBLEMS BY APPLYING ECONOMIC-BASED PRINCIPLES TO MOBILE AND WIRELESS RESOURCE CONSTRAINT NETWORKS

A dissertation submitted in partial fulfillment of the
requirements for the degree of
DOCTOR OF PHILOSOPHY
in
ELECTRICAL ENGINEERING
by
Concepción Zulema Sánchez Alemán

2020

To: Dean John L. Volakis
College of Engineering and Computing

This dissertation, written by Concepción Zulema Sánchez Alemán, and entitled Solving Complex Data-Streaming Problems by Applying Economic-Based Principles to Mobile and Wireless Resource Constraint 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.

Sundaraja Sitharama Iyengar

Kang K. Yen

Jean H. Andrian

Deng Pan

Niki Pissinou, Major Professor

Date of Defense: July 17, 2020

The dissertation of Concepción Zulema Sánchez Alemán is approved.

Dean John L. Volakis
College of Engineering and Computing

Andrés G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2020

DEDICATION

To my mother, Angela Argentina Alemán J., in memory of the dream we began together and she knows and sees, being in heaven. To my beloved daughter, Marianna Angela, my reason and strength. My dad and brothers: Ramiro Sánchez Reyna, Ramiro and Kedin Sánchez Alemán.

ACKNOWLEDGMENTS

First, I wish to express my deepest gratitude to my advisor, Dr. Niki Pissinou, whose guidance, encouragement and support has enabled me to complete this work. I am also thankful to her for her encouraging advice, knowledge and strict training helped me become an independent researcher, and overcome significant challenges.

I am thankful for all my thesis committee members: Dr. Sundaraja Sitharama Iyengar, Dr. Deng Pan, Dr. Jean Andrian, and Dr. Kang Yen for their time and knowledge shared while serving on my dissertation committee. Their comments and feedback were valuable to the completion of this work.

I also thank the excellent faculty members of the Department of Electrical & Computer Engineering and the School of Computing and Information Sciences for their lectures and projects. Thank you to Colonel Jerry Miller, Mrs. Patricia Brammer, Mrs. Olga Carbonell and Mrs. Ariana Taglioretti.

I want to acknowledge my lab mates at Telecommunication and Information Technology Institute (IT2). Especially thanks go to Sheila Alemany Blanco, for her valuable collaboration to my research. Also, thanks to Dr. Georges Kamhoua, Dr. Mingming Guo, Dr. Abdur Rahman Bin Shahid and Dr. Samia Tasnim for their constant encouragement and support.

I want to thank my family, especially my mother, Angela Argentina Alemán J for her unconditional love. Her hard work and perseverance has set an admirable example for me to follow. Moreover, I want to thank my aunts Modesta Alemán, Yara Suman and Doris Arroyo, for their support. Also, I would like to appreciate all the love and encouragement from John, Ruth, Luis and Lloyd during my dissertation.

I would like to acknowledge that my graduate studies have been partially funded by the National Bureau of Science, Technology and Innovation of the Republic of Panama (SENACyT), the US National Science Foundation, Department of Defense, the department of Electrical & Computer Engineering and the graduate school at Florida International University through the Dissertation Year Fellowship.

ABSTRACT OF THE DISSERTATION

SOLVING COMPLEX DATA-STREAMING PROBLEMS BY APPLYING ECONOMIC-BASED PRINCIPLES TO MOBILE AND WIRELESS RESOURCE CONSTRAINT NETWORKS

by

Concepción Zulema Sánchez Alemán

Florida International University, 2020

Miami, Florida

Professor Niki Pissinou, Major Professor

The applications that employ mobile networks depend on the continuous input of reliable data collected by sensing devices. A common application is in military systems, where as an example, drones that are sent on a mission can communicate with each other, exchange sensed data, and autonomously make decisions. Although the mobility of nodes enhances the network coverage, connectivity, and scalability, it introduces pressing issues in data reliability compounded by restrictions in sensor energy resources, as well as limitations in available memory, and computational capacity.

This dissertation investigates the issues that mobile networks encounter in providing reliable data. Our research goal is to develop a diverse set of novel data handling solutions for mobile sensor systems providing reliable data by considering the dynamic trajectory behavior relationships among nodes, and the constraints inherent to mobile nodes. We study the applicability of economic models, which are simplified versions of real-world situations that let us observe and make predictions about economic behavior, to our domain. First, we develop a data cleaning method by introducing the notion of "beta," a measure that quantifies the risk associated with trusting the accuracy of the data provided by a node based on trajectory behavior similarity. Next, we study the reconstruction of highly incomplete data streams. Our method determines the level of trust in data accuracy by assigning variable "weights" considering the quality and the origin of data. Thirdly, we design a behavior-based data reduction and trend prediction technique using Japanese candlesticks. This method reduces the dataset to 5% of its original size while preserving

the behavioral patterns. Finally, we develop a data cleaning distribution method for energy-harvesting networks. Based on the Leontief Input-Output model, this method increases the data that is run through cleaning and the network uptime.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

The emergence of Mobile Wireless Sensor Networks (MWSN) changed the attention from the common static wireless sensor networks to networks in which nodes are mobile. This mobility in the sensor nodes enables the possibility to achieve a world in which networks are pervasive and ubiquitous. As a result, the study of MWSN has taken an encouraging direction and their applications extend to a wide range of domains, including tactical military systems, homeland security, health care systems, environmental monitoring, vehicular systems, logistics, and industrial monitoring [YLL+14]. For example, in vehicular systems, autonomous vehicles that follow a common leader, separated by small inter-vehicle gaps, can form road trains to improve vehicular flow and reduce accidents [TCS17].

Typically, these types of networks consist of a large number of sensor nodes deployed over a wide area, where sensors share information, collaborate to perform operations, and autonomously make decisions [Ma11]. In essence, sensors enable the collection of information about the physical world to be utilized in data analysis processes for decision-making applications. The mobility of nodes facilitates the expansion of the coverage of networks and its rapid scalability. Nevertheless, the mobility also increases the difficulty in preserving data reliability due to data collision [SBB13a], sensor isolation, and short-term connectivity of the network [PGWC16]. Moreover, sensor nodes are expected to function properly for long periods of time and to provide reliable and accurate data, a critical factor in system functionality and reliable real-time decision-making.

In consideration of the data-centric nature of real-time decision-making applications, it is of remarkable importance to build resilient mechanisms to prevent these applications from making erroneous decisions. As expressed in November 2019 by Lt. Gen. Jack Shanahan from the Department of Defense, sensing data holds very high value as it is employed in crucial operations that go from preventive maintenance to targeting. Clean, accurate data helps to make military operations more

efficient, reduce collateral damage, and bring the military personnel home safely [gen19]. Given the relevance of data quality in MWSN, in this dissertation we investigate the challenges that MWSN confront when seeking to provide accurate and reliable data to employ in critical decision-making for real-time applications. We built techniques to handle sensor data streams employing economic theories.

Specifically, we investigate the application of economic models that simplify real-world problems, allowing us to observe, understand and make predictions about an economic behavior, to mobile and wireless network systems. Economic models originate from the examination of data, the individuals generating these data and the factors affecting their behavior. An economic model supports in the identification of correlations, and helps us to explain the causality behind these correlations. Once an economic model has been constructed, an economic theory is used to hypothesize the future data behavior and test if in fact these predictions are reflected in the data. Economic models and mobile networks scenarios share similarities including limited resources and the rationality of its participants. Moreover, the simplified view of economic models supports the development of methods with reduced computational complexity, a desirable characteristic for in-network processing. It is for this reason that in this research, we propose the application of economic theories to solve problems in mobile networks.

1.2 Motivation

According to the Allied Market Research, the global market of sensors was valued at $138,965 million in 2017 and is projected to reach $287 billion in 2025 [Res20]. The key element driving this exponential growth is its critical role in IoT applications, as these applications permit the collection of information about the physical world employed in data analysis for decision-making applications. MWSN are essential elements of the Internet of Things (IoT) as they increase the coverage of the Internet and the expansion of computing [YH18]. Due to the increasing adoption of MWSN, extending its life to continuously collect real-time and reliable information has become of crucial interest. Nevertheless, in MWSN the two greatest energy con-

sumers are computational operations in in-network processing and communication tasks [AQAKS17]. Existing data handling methods rely on the presence of a sink or a base station for their data processing, and most of them do not consider the sink isolation that leads to network failure, network delays caused by the transmission of large volumes of data to a sink for processing, and energy holes caused by the energy exhaustion in sensors near sink due to heavy traffic. All these difficulties result in high volumes of missing, noisy or duplicated data in MWSN [DWW+19], and this data imprecision can lead to erroneous decisions. For example, in the military, deploying soldiers can be risky and dangerous. It is for this reason that autonomous mobile nodes can be deployed to patrol hostile territories to gather and distribute information to be employed in different applications including perimeter surveillance and protection, nuclear, chemical, and biological attacks detection, and missile monitoring [AFS17]. If sensors fail to provide accurate data in the observed environment, soldiers can be commanded to proceed to the dangerous area, and this decision can lead to unnecessary troop fatalities.

Moreover, although real-time monitoring and prediction of future data values are beneficial for decision support, the projection of data behavior trends is particularly relevant for applications that seek to take preventive actions. While the prediction of specific values can assist in taking preventative measures, the ability to foresee the direction of the data evolution may have the same impact without the added computation involved in the prediction of data values. Two applications that directly benefit from the data behavior analysis and prediction of data trends are environmental monitoring and remote health care monitoring. In environmental monitoring, air quality is a significant problem for public health, particularly in metropolitan cities [HAdC+15]. In 2012, the World Health Organization (WHO) approximated that air pollution produced 3 million premature deaths each year, and by 2016, these deaths increased by 40%. This fatality is a result of the exposure to small particles, which can cause heart disease, lung disease, and cancer [O+18]. In 2016, more than half of the world population was living in places where the air pollution levels in the outdoors were at least 2.5 times above the safety standard set by WHO. The use of mobile nodes and IoT devices enables cities to rapidly expand

their monitored area without incurring excessive costs of network infrastructure deployment. Mobile sensor nodes embedded in vehicles and hand-held devices can collect air quality data within citizens' trajectories throughout the cities. The rise in pollution levels can be predicted to take preventive actions and limit the exposure to highly polluted air, thus impeding massive health issues. In remote health care monitoring applications, patients can carry wearable sensors that communicate with IoT devices. Health care providers can collect vitals information to facilitate the analysis of a patient's condition for diagnostics and preventive/emergency treatment decisions. General medical practice utilize a numeric thresholds to act towards a specific patient. In other words, a patient's health condition needs to reach a certain level of severity before medical intervention. The discovery and prediction of patterns in a patient's vitals data evolution can help to determine the patient's condition promptly, before surpassing a pre-set threshold. This prompt condition discovery can provide additional time to attempt to counteract the deteriorating state of the patient and potentially save the patient's life.

These examples motivate our research in new methodologies to mitigate the negative effects that mobility and resources scarcity impose over these types of networks. The development of light-weight methods that seek to reduce energy consumption while providing highly accurate data is necessary. Therefore, different data handling methods to ensure the availability of reliable data in MWSN have been proposed and evaluated in this research. The aim is to eliminate unnecessary energy expenditure in order to extend the nodes' lifetime without having to trade off data quality.

1.3 Research Problems

The aforementioned examples demonstrate the importance of addressing the challenges that mobility and resources constraint inflict in real-time applications that are dependent on data collected by sensor nodes. The main problem that this dissertation undertakes is that nodes composing wireless and mobile networks are have limited resources and that differently from static wireless networks, the availability

of a base station to support heavy computational operations is unrealistic. Firstly, it means that existing data handling techniques that rely on a base station or a sink are not available for mobile scenarios. Secondly, it has been observed that in MWSN's dynamic environments, nodes may have only one interaction with specific nodes. It is for this reason that trust in data accuracy must be determined quickly. Effective data handling methods in MWSN are required to contemplate the mobility of sensors. Even though previous research studies have pursued better data quality in static networks, there is still much work needed to improve data quality in MWSN. The objective in [GLN15] [LBX+16] [KSY+14], [ZSS15], is to clean dirty data, and although [LBX+16] showcased effectiveness when tested with various real datasets, authors only considered static environments. While [KSY+14], [ZSS15] and [GLN15] were proposed for wireless environments, due to its high computation the presence of a sink or a base station is still required for processing, making it unsuitable for completely mobile scenarios.

On the other hand, most data trend prediction techniques employ computationally heavy methods, including Neural Networks and machine learning [BM16], [WY17],[WTL+17], [GSB+18]. However, the energetic cost assumed by computational operations in these methods is too expensive to be performed at the sensor node. Also, sensor nodes are still required to spend an ample amount of their energy to transmit the sensing data to the base station. Furthermore, complex abstraction processes, including Principal Component Analysis (PCA), have been implemented in [WTX16], [FK17], [IUK16], but these methods are also very expensive to be performed at the sensor node level.

Finally, in an effort to reduce the energy constraint in MWSN, energy-harvesting technologies have been applied. Although the sensor's lifetime is not a problem in energy-harvesting mobile wireless sensor networks (EH-MWSN), when the onboard residual energy of a node goes below a pre-set threshold, the sensor adopts an energy saving strategy and becomes inactive. Once it has harvested enough energy it becomes active again and to reassume its normal operations. These energy saving strategies reduce the sensor's functionality including its ability to sustain data reliability. To reduce network downtime, substantial attention has been placed in

methods involving the management of energy [FD11, GHZH14, ZCZ+16, KHZS07, VGB07, ZSA11, SSCS17, TAH+15, Cui18]. Nevertheless, limited attention has been placed on increasing the quality of the sensed data. With the large quantities of dirty data provided by mobile nodes, data cleaning becomes critical due to the negative effects that dirty data have over data mining, machine learning models and other techniques employed in decision-making applications [QWLG18, PGWC16]. While these methods use diverse approaches to manage the power used in communication, sensing and data processing in EH-WSN, the mobility of nodes and the variability in energy availability add complexity to the challenges already present in static networks. In the ideal EH-MWSN the network performance is maximized while sustaining a harvesting rate higher than the energy expenditure rate, and sustaining this goal depends on the energy harvested by multiple distributed nodes. Undoubtedly, there are still problems that need to be addressed to handle data in MWSN.

1.4 Research Objectives

In MWNS, sensor nodes are constrained by low memory, computational capacity and limited energy resources. Also, the mobility of nodes promotes the high dynamicity of its network topology. These limitations make it difficult to ensure reliable data for decision-making applications. This research stems from the realization that the application in MWSN will not be exploited to their highest capabilities unless methods that take into consideration sensors' resources constraint and mobility into data handling mechanisms are developed. This dissertation includes the design of methodologies and evaluation results of different scenarios that require data handling to ensure reliable data for real-time decision-making applications in MWSN. Specifically, we investigate the following four topics:

Diversification of Trust to Clean Data in MWSN

Selecting a sensor node to support in the data cleaning processes is a primary challenge that is not well tackled in existing sensor data cleaning methods. In par-

ticular, 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 MWSN. Even though there are existing techniques that take advantage of the Spatio-temporal characteristics exhibited in mobile environments, other factors can affect the sensed data collected by sensors. We hypothesize that we can select the most helpful set of neighboring sensors to support the data cleaning process of a sensor if we evaluate a set of parameters to measure the trustworthiness of sensor data accuracy based on trajectory behavior similarity using economic theories. At the same time, we can minimize the error in data estimation by diversifying the risk associated with trusting data accuracy among this set of selected nodes. The greater the trajectory behavior similarity, the easier a set of trustworthy sensors can be selected. Therefore, our first objective is to develop a data cleaning method to model trust in data accuracy based on the trajectory behavior similarity of sensors in a pre-defined area.

Dynamically Allocate Trust Weights to Reconstruct Data in MWSN

According to the developed method for trust diversification, we found that MWSN exhibit different types of data loss patterns [KXL+13]. Methods that consider a combination of these data loss patterns can help to reconstruct highly incomplete datasets. Additionally, we found that due to the mobility of sensors, they may have been close to each other during the data collection period, but may never come close again, making it difficult to find a set of sensors that can provide all the information required to clean data. We hypothesize that we can determine the trust level in the data accuracy of each candidate node in MWSN using economic theories. In this type of networks sensors experience data loss due to noise and collision, unreliable links, sensors losing energy or malfunctioning. Therefore, the objective of this task is to develop a data reconstruction method capable of evaluating second-hand data when there is no first-hand data available. This method should also select second-hand data when this data is more accurate than the first-hand data by assigning variable "weights" considering the quality and the origin of data

Modeling of Data Behavior and Data Trend Prediction for MWSN in IoT

The methods that seek to predict future data trends take advantage of IoT technologies to perform their computational operations. However, the massive loads of data needed to be transmitted from the sensor node to IoT devices are excessive and the energy depletion problem still prevails. Moreover, existing data reduction methods are either too complex to take place at the sensor node, or they fail to represent the behavior of the real phenomena being observed. We hypothesize that the use of economic theories to extract the features that describes the behavior of individual sensor's collected data can effectively model data behavior in MWSN. Therefore, this part of our research is directed to develop an effective data behavior modeling method using sensor's historical data. The output from this data behavior modeling analysis can be employed in predicting the future data behavior trend.

Data Cleaning Workload distribution in EH-MWSN

The energetic cost of data cleaning can be elevated for sensors with large loads of dirty data. Although energy-harvesting-enabled nodes promise to deliver infinite lifetime when deployed in environments with a constant energy supply, the heterogeneity and mobility of the sensors add challenges that severely affect the collection of high-quality data [SBB13b] and uptime extension. Our hypothesis is that the distribution of the data cleaning workload in EH-MWSN powered by predictable energy sources can increase network uptime and the quantity of data that is run through data cleaning processes. Therefore, the aim of this method is to ensure the availability of accurate data and increasing the uptime in networks composed of nodes possessing heterogeneous functions and capabilities.

1.5 Research Contributions

In this dissertation, we investigate the challenges that MWSN encounter when handling data. According to the research objectives, we focused on developing diverse data handling solutions that include protocol design, algorithm development, exper-

imental and simulation results and analyses. These solutions involve (1) developing a data cleaning method that models trust in data accuracy based on the similarity of the continuous evolving trajectory and Spatio-temporal relationships of moving sensors; (2) building a data reconstruction method that evaluates second-hand and first-hand data accuracy in highly incomplete datasets; (3) creation of a data behavior modeling that reduces the space complexity in a mobile node and that can that can be employed in accurately predicting the data behavior trends; (4) distributing data cleaning workload in EH-MWSN to increase the availability of reliable data and extend network survivability. Specifically, we make the following contributions.

Cleaning Dirty Data in Mobile Wireless Sensor Networks [SAPA+18]

Contrary to previously mentioned research works [LBX+16, GLN15, KSY+14, ZSS15], we address the problem of selecting a set of sensors to support during data cleaning in mobile environments where the network topology is dynamic. Particularly, we select this set of sensors by measuring the trustworthiness of sensor data accuracy based on the similarity of their trajectory, and their Spatio-temporal relationship. This method compares (i) the similarity in trajectory behavior of each candidate node with respect to a baseline node, (ii) calculates the spatial autocorrelation of neighboring nodes, and (iii) minimizes the error in estimating the dirty data sample by diversifying the trust in data accuracy among the selected nodes.

The major contributions we have made with this work can be summarized as follows: this work introduces a unique data cleaning method tailored for dynamic mobile environments where sensor nodes are connected for short periods of time, so they need to quickly determine which received data has the highest accuracy. We present an economic-based approach to identify the trustworthy set of sensors to support during the data cleaning. Our method assigns trustworthiness weights to the selected candidate sensors, utilizing the computation of two Beta scores, the Speed Beta (β_s) and the Angle of Travel Beta (β_θ). The combination of these Beta scores measures the trajectory behavior similarity between two nodes [Section 3.3.1]. Then, we select the set of candidate sensors that will support the process of data cleaning employing local Moran's I. Local Moran's I separates clustered sensor nodes

from outliers by computing the spatial autocorrelation, among the group of previously selected sensor nodes [Section 3.3.2]. Additionally, to estimate the missing data we assign weights to the different candidate sensors, based on the risk to trust each sensor's data [Section 3.3.3]. Our results show that samples cleaned by the proposed method exhibit lower percent error when compared to other well-known and effective data cleaning algorithms in tested outdoor and indoor scenarios. This content was published during my Ph.D. study. [1].

Reconstructing Highly Incomplete Data in Mobile Wireless Sensor Networks [APAK18a]

To overcome the limitations of the aforementioned data cleaning method, we focused on investigating how to reconstruct incomplete data in mobile networks where there is not first-hand data available or the second-hand data available is more accurate that the first-hand data. This method (i) evaluates the number of correct observations provided, (ii) prioritize first-hand data and consider second-hand data using the Euclidean distance between two nodes, (iii) quantify the strength of a linear relationship between the collected sensed data, and (vi) measures the trajectory similarity between a pair of sensors.

The major contributions we have made with this work can be summarized as follows: this work proposes a light-weight data reconstruction method for mobile environments, where energy preservation is crucial. We present a novel data reconstruction method to identify a set of sensors to support the prediction of missing data. Our method dynamically assigns weights for trust in data accuracy to first-hand and second-hand data in highly incomplete data without the usage of a predefined threshold. Our proposed scheme describes trustworthy nodes as nodes containing the highest quantity of high-quality, spatiotemporally correlated data with resemblance in trajectory behavior in relation to the evaluating node.

---

[1] © [2018] IEEE. Reprinted, with permission, from [Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, Kianoosh Boroojeni, Jerry Miller, Ziqian Ding, Context-Aware Data Cleaning for Mobile Wireless Sensor Networks: A Diversified Trust Approach, 2018 International Conference on Computing, Networking and Communications (ICNC)]

To compute the total trust, our method evaluates a set of parameters including: confidence level, Spatio-temporal closeness, the Pearson's Correlation coefficient, trajectory similarity [Section 4.4.1]. Lastly, the total trust score is computed by combining the evaluated parameters [Section 4.4.2], and the missing values are estimated using trust diversification based on the number of available trustworthy sensor nodes [Section 4.4.3]. Our results demonstrate that data reconstructed using our dynamic trust allocation method depicts a significant lower Root Mean Square Error (RMSE) compared to methods that only consider Spatio-temporal and sensed data values correlation. Our approach showed consistent outstanding performance by achieving high data accuracy in datasets containing vast quantities of missing data. This content was published during my Ph.D. study. [2].

Modeling Data Behavior and Predicting Trends for Mobile Wireless Sensor Networks in IoT [APAK18b]

For scenarios where the prediction of the trend of the data behavior can help to take preventive actions, we develop a method to model the behavior of the sensed data and predict the future trend of the data. We perform these predictions using the value that describes the behavior of the data from a time partition to the subsequent one as input for the SVM. The major contributions we have made with this work can be summarized as follows: we employ Japanese candlestick data abstraction to extract the main features of data in the different time partitions [Section 5.4.1]. Also, we use this abstracted data together with dynamic time warping to model sensors' data evolving behavior in real-world applications of MWSN in IoT [Section 5.4.2]. This data behavior modeling reduces overall space complexity. Also, we utilize a supervised learning algorithm, multi-class SVM, to accurately predict sensor nodes' future data trends [Section 5.4.3]. A comparative study was conducted

---

[2]

to investigate the effectiveness of our method on real-world datasets. Our results show that data trends predicted achieve better precision, recall, and accuracy score when contrasted against four well-known techniques while reducing the space complexity by at least a factor of 10. This content was published during my Ph.D. study. [3].

Distributing Data Cleaning Workload in Energy-Harvesting Mobile Wireless Sensor Networks[SAPA20]

In EH-MWSN, when a node adopts energy-saving strategies, its ability to sustain data accuracy gets limited. It is for this reason that we develop a method designed to distribute the data cleaning workload in energy harvesting MWSN. This method (i) creates interrelations between sensor nodes and (ii) distribute the data cleaning workload among these nodes. This procedure reduces the downtime of nodes and increases the quantity of data that is run through data cleaning processes.

The main contributions made with this work can be summarized as follows: this work proposes an economic-based data cleaning workload distribution method that employs sensors' current and predicted onboard residual energy to compute a data cleaning workload distribution strategy that seeks to achieve Neutral Network Operation, a state in which the energy harvesting rate of the network is greater than its energy consumption rate [Section 6.4.1]. Our method increased network survivability by planning this workload distribution considering the network as a whole, rather than only individual sensors, which consequently benefits the overall system performance [Section 6.4.2]. We evaluate the performance of our proposed method using real-world datasets. The results show that our data cleaning workload distribution method increases the number of data samples engaged in data cleaning processes by up to 25.57%. This technique also increases the count of active sensors by up to 44.01%, and the overall well-being of the network by up to 55.42% when

[3]

compared to data cleaning performed by each sensor node individually. This content was published during my Ph.D. study. [4].

1.6 Organization of the Dissertation

The outline of this dissertation is as follows: Chapter 1 presents an introduction to the research in this dissertation. It describes the background, challenges, research objective, and the overall contributions of this dissertation. In Chapter 2, we review a comprehensive literature related to data handling methods employed in MWSN. Chapter 3 presents the diversified trust approach to clean data that analyzes the behavior trajectory similarity of sensor nodes to select the set of sensors to support the data cleaning process. We describe the dynamic trust weight allocation method to reconstruct incomplete data in Chapter 4. Chapter 5 covers the data behavior modeling and data trend prediction and Chapter 6 discusses the data cleaning workload distribution strategy and its justification. Lastly, in Chapter 7 we conclude what we achieved and provide a conclusion and recommendations for potential further research directions.

---

[4]

CHAPTER 2

RELATED WORK

Applications in MWSN require a continuous input of data streams to be employed in data analytics and real-time decision-making. Having reliable data is important, as inaccurate or incomplete data can lead to unfavorable outcomes. Unreliable data refers to noisy data, missing data, duplicated data, etc. Existing data handling methods that seek to preserve data accuracy for applications in MWSN include data cleaning, data reconstruction, data abstraction, and data classification. Nevertheless, the mobility of nodes in the network increases the difficulty in preserving data accuracy due to energy, memory and computational power constraints.

The aforementioned constraints prevent the existing methods designed for static WSN from being employed in mobile environments. The purpose of this chapter is to provide a brief literature review on data handling methods in static WSN and MWSN. We organize and present each section based on the data handling method.

2.1 Data Handing Approaches

2.1.1 Missing Data Estimation

Multiple data cleaning techniques have been proposed to improve data quality. In [CFSC18], Cheng et al. presented a quality-based data cleaning method. This method employed a quality assessment by evaluating the relationship between sensors' data quality indicators. Then, the results of the quality assessment are used to propose a sequence in which data cleaning needs to be carried over. Also, Zhang et al. presented a reliability-based technique, where the reliability of each sensor is adapted according to its performance [ZSS14]. At every iteration, the consistency is updated based on the difference of the prediction made and the real sensed value. These selected sensors' data was used to improve data quality in environmental monitoring. Later, a method for selecting a reliable sensor was presented by Zhang et al. [ZSS15] using a statistical model based on sensed data and latent variables, such as the sensors' faulty state. On the other hand, Ghorbel et al. [GASA15] detected outliers using Mahalanobis distance (MD) and the kernel principal component analysis

(KPCA). The authors separated outliers from the normal data distribution patterns by computing mapping data points and then mapping the data to another feature space.

Moreover, many techniques have been proposed for predicting missing data. In [KXL+14], Kong et al. proposed Environmental Space Improved Compressive Sensing (ESTICS). ESTICS employed compressive sensing while combining Spatio-temporal correlation to reconstruct complete information from a portion of data. Later, Lei et al. [LBX+16] estimated the missing values in incomplete sensor data by repeating two processes: selecting spatially correlated sensors and updating the training sensor dataset with the data selected from those sensors. This procedure assists in obtaining a more suitable neighbor sensor and refines the regression model by querying a record within the training sensor dataset. Chen et al. proposed an environmental data reconstruction method based on a guided temporal stability matrix [CCH+18]. The method employed the block coordinate descent method and the operator splitting technique. Additionally, the accuracy in the reconstructed data was increased by introducing a constraint about short-term stability to the matrix completion that enabled the erroneous data recognition.

Considering the dynamic nature of mobile environments, other methods were proposed for MWSN. In their research, Gill et al. proposed a context-aware model-based technique for cleaning environmental data from sensors [GLN15]. The authors used geographical and meteorological datasets to create statistical models to train the system. Outliers are identified and discarded, then the partially cleaned data is analyzed by comparing the observed value with the predicted value for each attribute. Every time the observed value surpasses the error threshold concerning the predicted data, the predicted data is used to replace the observed value. Furthermore, multivariate linear regression was employed by Kurasawa et al. to predict the missing data by exploiting multiple attribute correlation [KSY+14]. The method exploited Spatio-temporal relationships and used machine learning to build training datasets through the back end, sending data back to the sensor periodically. Later, Tasnim et al. [TPI17] proposed a data cleaning technique for ensuring data accuracy for applications of MWSN in environmental sensing. The authors selected a sensor

to help during the data cleaning taking into consideration the mobility pattern of the sensor nodes. In this work, sensors computed and updated a credibility value of the historical sensing performance of sensor nodes and a context credibility value using the context value of neighboring sensors during the time window.

Although [KXL+14], [GLN15], [ZSS15], [ZSS14], [GASA15] and [KSY+14] exhibited high levels of effectiveness, the importance of methods that perform in-network computations and capable of handling large volumes of trajectory data is not considered. Moreover, the energy consumption to perform the heavy computations that most of these models propose is elevated. It is for this reason that these methods are not transferable to MWSN. Light-weight techniques are crucial, as sensors deployed in mobile environments tend to work unattended with limited power and computational capacity [FZ16]. Despite the fact that [KXL+14], [LBX+16], and [ZSS15] were proposed for wireless environments, and [GLN15] and [KSY+14] for mobile networks, they relied on the presence of a sink and/or a back-end for their data processing. MWSN tend to present delays related to continuous data transmissions and delays related to the data processing even when there is no data to reconstruct, as in [ZSS14]. Also, authors in [KXL+13] did not consider network failure due to sink isolation produced by the sensors near the sink that deplete their energy faster due to heavy traffic, creating energy holes and network failure.

2.1.2 Future Data Prediction Methods

The prediction of future data was proposed by Wu et al. [WTX16]. In this method, sensor nodes selectively sent data to a cluster head. The cluster head then made predictions based on the received readings using the least mean square prediction algorithm but added an adaptive optimal step size parameter that minimized the mean square derivation. Data features were reduced using the Principal Component Analysis (PCA) and sent to the base station. Upon receiving the data, the original values were recovered for further calculations. Barton et al. [BM16] proposed a method where neural networks were employed to predict future trends. The model was generated by utilizing the calculated slope of a linear fit that started in the

current time and ended in the future. This method claimed to reduce the number of model updates while predicting future trends.

These methods were employed to predict future sensing samples and may reduce the energy expenditure related to sensing and data communication tasks in WSN [WTX16, BM16]. However, the energy used in computational operations by neural networks and complex data feature analysis, such as PCA, are too expensive to be performed at the sensor node. Also, sensor nodes are still required to spend an ample amount of their energy to transmit the sensing data to the base station.

On the other hand, there are many techniques explicitly used to take preventive actions in multiple diverse fields. In the health care field, Forkan et al. [FK17] employed PCA to extract and validate variations of different vitals data. It clustered normal and abnormal states. These observed behaviors were passed into a Hidden Markov Model (HMM) to conduct the prediction of various vitals as a sequence of temporal dependent time series. Ghazal et al. [GSB+18] proposed the use of machine learning methods to predict hemoglobin oxygen saturation levels (SpO_2) five minutes after a ventilator setting changed. The authors classified the saturation levels and balanced the data so that the classifier learned the majority class labels equally. The method predicted SpO_2 classifications employing an artificial neural network and bagged complex decision trees. Ravishankar et al. [**?**] proposed a pattern detection algorithm to identify respiratory distress in hospitalized patients. The technique employed temporal abstractions followed by a Markov Model-Based Finite State Machine to predict the condition before the violation of the SpO_2 acceptable threshold.

In the industrial field, Wan et al. [WTL+17] proposed a method for preventive maintenance in a manufacturing environment using neural networks. The technique depicted real-time active support to fulfill the real-time requirements of operations and an off-line prediction and analysis to forecast failures in the different components. Wang et al. [WY17] also designed a real-time data monitoring framework to ensure food quality in the supply chain network and prevent food recall. It employed an association rule mining and IoT technology to monitor and share data among all agents involved in the supply chain. The framework predicted food safety

risks to give decision-support information to maintain the quality and safety of food products. While these methods seek to predict future data trends take advantage of IoT technologies to perform their computational operations, the massive loads of data streams needed to be transmitted from the sensor node to IoT devices are excessive and the energy depletion problem still prevails.

2.1.3 Classification-Based Prediction Methods

The general usage of classifiers in WSN has also been studied. Islam et al. [IUK16] presented a technique to diagnose faults in electric motors employing PCA for features abstraction and a multi-class SVM for classification and training of different types of faults at the sensor node. Samanta et al. [SBS16] employed K-Nearest Neighbors (K-NN) to diagnose faults in the same type of motors in an online fashion. Both employed sequence component analysis. The positive and negative sequence components were computed utilizing Sample Shifting Technique. K-NN was then used to diagnose a faulty phase and severity of the fault. Patel et al. [PG16] proposed a multi-class fault detection and diagnosis method for condition-based maintenance for bearings in rotational machinery using the Random Forest classifier. They applied a statistical parameter extraction from the sensed vibration data and used it as an input feature for the classification task.

Elghazel et al. [EMZ+15] proposed a method for industrial devices functioning diagnostics using Random Forest in the presence of data streams with a heterogeneous number and quality of features. In [RMH+19], Rida et al. proposed EK-Means for reducing the redundancy in data. The proposed method was divided into two levels In the first level, the sensor node collected data and cluster it based on the Euclidean distance. In the second level, the intermediate node aggregated the data received from the neighboring node. Later this data was clustered one more time based on their spatial correlation. Even though the aforementioned classification techniques were effective in accurately diagnosing faults in WSN, none considered a data abstraction technique to reduce the complexity at the node level. Authors did not consider scenarios that included mobile nodes and where light-weight data

reduction methods are crucial to avoid energy depletion and prolong network functionality.

2.2 Energy saving Strategies Energy-Harvesting Networks

The use of predictable renewable energy resources to power sensor nodes is a promising approach toward achieving self-sustainable MWSN. However, the uncertainty in the availability of this type of energy requires the implementation of techniques to manage sensors' resources and ensure the successful completion of tasks. For this reason, research work has been directed to employ energy availability prediction to adjust the nodes' operations to satisfy Energy Neutral Operation (ENO), a state in which a sensor node harvests more energy than it consumes. In recent work, Cui et al. [Cui18] used long short-term memory recurrent neural network (LSTM-RNN) to predict solar energy in the subsequent days using solar energy historical data together with environmental data. Next, the method carried out a predictive task-scheduling strategy based on the predicted energy.

Additionally, Kansal et al. [KS03] proposed a dynamic duty cycle method that kept a summary of the energy generation and employed Exponential Weighted Moving Average (EWMA) to predict future energy availability for harvesting. The duty cycle was reduced or increased based on the required energy and the actually harvested energy [KHZS07]. Moreover, Vigorito et al. [VGB07] adapted the duty cycle to achieve ENO in WSN utilizing adaptive control theory. This model-free method sought to provide a stable duty cycle in environments under dynamic conditions by controlling the energy supply level parameter.

Furthermore, Fafoutis et al. [FD11] presented an On-Demand Medium Access Control method to achieve ENO state. The energy-harvesting rate and battery level were used to dynamically adjust the duty cycle by computing the duration of the sensing period. When a node was available for reception, it broadcast a beacon packet to alert nodes needing to transmit. It then employed an opportunistic forwarding scheme to reduce the energy wastage caused by long wait time. Subsequently, Tan et al. [TAH+15] proposed a topology control strategy. The authors

used game theory to model the behavior of sensor nodes as an ordinal potential game with an existing Nash equilibrium. This method considered sensors' energy status and harvesting capabilities to encourage cooperation among the high and low harvesting power nodes to optimize the network topology. Finally, the transmission power of the node is determined by analyzing the rates of consumed and harvested power.

In [SSCS17], Shu et al. presented a utility-based sensing rate allocation algorithm that exploited the redundant deployment of sensors. It computed the optimal energy replenishment and coverage control strategy to achieve a harvesting-aware task scheduling. Likewise, Zhang et al. [ZSA11] proposed a method to maximize the total application utility. This sensing rate allocation epoch-based algorithm defined the utility of a node based on its packet rate. This method used a collection tree protocol to organize sensor nodes as a data collection tree. Moreover, Gu et al. [GHZH14] synchronized nodes' activity patterns with the available energy budget. The method used an energy synchronized communication protocol (ESC) to reduce data forwarding delays and to increase data delivery using the excess harvested energy, rather than maximizing conserved energy subject to leakage. Additionally, Zhang et al. [ZCZ+16] proposed a cooperative transmission scheme that balanced the node's residual energy. The method chose the sensor with the maximum residual energy to cooperate as a relay for a source node based on the initial energy in this source, its energy harvesting rate, and the channel gain.

While the above methods use diverse approaches to manage the power used in adjusting sensor nodes activities such as communication, routing and sampling based on their energy harvesting and consumption rates in WSN, the mobility of nodes and the variability in energy availability add complexity to the challenges already present in static networks. Moreover, data accuracy is still a critical component in MWSN applications, and given the exponential growth of sensing data that is being generated, it is important to consider the development of methods that seek to increase data reliability without affecting network uptime.

2.3 Summary

In this chapter, we have reviewed existing approaches to handle data and static and mobile wireless sensor networks. The work discussed above magnifies the importance and the need for the development of data handling methods designed to solve the problem of data reliability in real-time applications in MWNS. Furthermore, we discussed energy-saving strategies and their role in ensuring data accuracy in EH-MWSN. In the following chapters, we will present the approached we developed to solve some of the open problems, the results of our evaluations, conclusions and future research directions.

CHAPTER 3

DATA CLEANING FOR MWNS: A DIVERSIFIED TRUST APPROACH

Dirty data is a prevailing problem in mobile wireless sensor networks. The mobility of sensor nodes challenges the data cleaning process in the existing methods employed in static wireless sensor networks. In this chapter, we address the problem of identifying a set of sensor nodes with the most accurate data to help during the data cleaning in mobile environments. This method introduces a novel economic-based approach to quantify risk in trust data accuracy to later diversify this trust and minimize the error in cleaning the dirty sample. Our method has the ability to effectively clean more than 92% of the dirty data under 5% error threshold, outperforming existing well-known methods. This chapter is organized as follows: An introduction about data cleaning in MWSN is presented in Section 3.1. The problem statement is reviewed in Section 3.2. The proposed method is introduced in Section 3.3. The evaluation of results is described in Section 3.4. Lastly, a summary of this chapter is presented in Section 3.5.

3.1 Introduction

The Diversified Trust Portfolio (DTP) proposed in this chapter employs the calculation of "beta" to measure the trajectory behavior similarity between two nodes. In essence, beta analysis allows for a comparison of trajectory behavior of each candidate node with respect to a baseline , determining the set of sensors with the most accurate data to clean the dirty samples. In addition to the introduction of betas, this technique combines the use of diversified portfolio, form Modern Portfolio Theory (MPT), to find an effective trust distribution. Diversified trust portfolio computation seeks to minimize the error percentage between the value of reference and the predicted value (product of our approach). Both concepts, betas and portfolio diversification, are introduced in the Capital Asset Pricing Model (CAPM) in financial-field applications [Dam], [Mar52].

The CAPM describes the relationship between the expected return of a given asset and the risk measured by beta coefficient. In finance, risk refers to the degree

of uncertainty and/or potential financial loss inherent in an investment decision. Similarly, in MWSN, we define risk as the degree of potentially selecting inaccurate/ unrelated data to be employed to estimate the data that will support in decision making applications, as inaccurate data translates into wrong decisions. Similarly to investors in the CAPM, mobile nodes are risk aversed and have access to all available information. We consider sensors tell the truth and provide all available data unselfishly. To the best of our knowledge, this is the first study that utilizes the beta calculation in combination with portfolio diversification from CAPM in modeling trust. Since the assumptions of CAPM resemble the assumptions made in our MWSN model, we can apply Modern Portfolio Theory (MPT), which states that a specific risk can be removed or at least mitigate through diversification

3.2 Problem Statement and Assumptions

The location of mobile sensor nodes changes dynamically over time; it is for this reason that sensor nodes cannot rely on a permanent set of sensors to help during data cleaning. Spatio-temporal characteristics have been employed in the past to evaluate the correlation among sensor nodes and clean the dirty samples. Nevertheless, it is our hypothesis that a set of sensor nodes can be selected to determine the trustworthiness of data accuracy by evaluating the trajectory behavior similarity of among sensor nodes. Figure 3.1 shows an example of a MWSN where sensor nodes embedded into vehicles and/or devices carried by people move in a determined pattern. At first glance, a group of sensors sharing similar locations may appear to have similar behaviors. But this may not be the case. For example, the speed at which the node with dirty data travels compared to the speed of the neighboring nodes may be a factor in determining the most trustworthy node. Moreover, selecting a single node can reduce the accuracy of the estimated values as the data of the selected node could be corrupted or imprecise. Selecting a set of candidate sensors can help to minimize the error during the estimation of the missing data. Once we have selected the set of most trustworthy sensors, the cleaning is performed using

their data to calculate and replace the missing values.

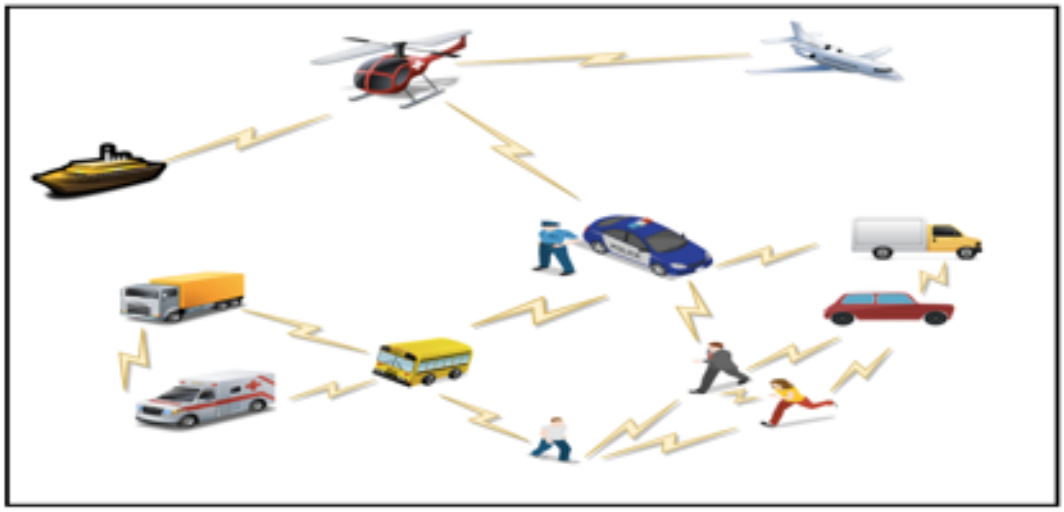


Figure 3.1: Example of a mobile wireless sensor network

We consider a decentralized, in-network computational method. This means the detection of dirty data, the selection of the nodes to clean data, and the data cleaning are all done by each sensor. We assume that each sensor node has an internal pre-process for detection of dirty data. Sensors are assumed to be (1) mobile, (2) to cooperate, and (3) to have *a priori* knowledge of the area where they are deployed. Also, each sensor node has a unique identity and its sensing task takes place asynchronously. The data exchange will only take place between any two nodes if these nodes are within transmission range. In other words, no data exchange will take place via multi-hop communication. Data exchanges follow the Round Robin Scheduling Technique as explained in [SJG15]. With the data from its surrounding sensor nodes, each node should be able to approximate the missing values. However, finding a trustworthy sensor becomes a challenge.

3.3 Diversified Trust Approach

Our method utilizes the computation of two beta scores, the Speed Beta (β_s) and the Angle of Travel Beta (β_θ), combined with the spatial autocorrelation, local

Moran's I, to select the set of candidate sensors for the process of data cleaning. The local Moran's I measures the spatial autocorrelation between a group of sensor nodes, while the beta computations compare the behavior of a group of sensors in relation to the node under analysis [YTW+16]. These computations quantify the risk involved in trusting a set of spatially correlated candidate nodes, as behavior similarities are directly proportional to their level of trustworthiness.

Data Gathering

Our data cleaning process requires a time window T which is partitioned into two phases: sensing phase T_s and cleaning phase T_c; at the same time T is divided into t time instants. Individual time instants are referred to as t_j where $j = 1, 2, ..., t$.

Sensing Phase (T_s)

When the sensor is performing the cleaning of its data, nodes with dirty data will select the fraction of time, T_e, within T_s to carry out the evaluation of candidate sensors. The time partition can be defined as: $T_e = t_d - k, ..., t_d - 2, t_d - 1, t_d, t_d + 1, t_d + 2, ..., t_d + p$, where t_d is the time instant where there is a missing value. The initial and final time instants in T_e are defined as $t_o = t_d - k$ and $t_f = t_d + p$ accordingly. $|T_e|$ is user-defined and p, k are dynamic values dependent on the time instant to be cleaned, t_d. For example, if the sensor needs to clean the first time instant in T_s, its T_e will only evaluate the time instances after t_d. Therefore, k and p are assigned 0 and $|T_e| - 1$, respectively. If t_d is not near the initial or final time instances in T_s, $p = k = \frac{|T_e|-1}{2}$. At the end of every t_j sensors will update their internal tables with the sensed value x_{ij}, its position $(x, y)_{ij}$, angle of travel θ_{ij}, and average speed s_{ij}, where $i = 1, 2, ..., n$, and n is the total number of sensors in the network. To avoid error propagation, once a sensor node detects a dirty sample within its sensed values, the value is marked with a flag. No dirty data will be considered during the process of data cleaning. Table 3.1 shows the internal table for sensor i at each t_j during T_s.

Time t_j	1	2	...	j
Position $(x, y)_{ij}$	$(x, y)_{i1}$	$(x, y)_{i2}$	...	$(x, y)_{ij}$
Sensed Value x_{ij}	x_{i1}	x_{i2}	...	x_{ij}
Average Speed s_{ij}	s_{i1}	s_{i2}	...	s_{ij}
Angle of Travel θ_{ij}	θ_{i1}	θ_{i2}	...	θ_{ij}
Dirty Flag f_{ij}	f_{i1}	f_{i2}	...	f_{ij}

Table 3.1: Sensor i Internal Table for T_s

Cleaning Phase (T_c)

The cleaning phase begins after the sensing phase. In this period if sensors have data to be cleaned, at every time t_j, sensors will search for neighboring nodes within transmission range by broadcasting a cleaning status message (CSM) containing its identification and current location. After the CSM has been received, sensors estimate if there is enough time to send its data to neighboring sensors before the neighboring sensors move out of transmission range, as in [NP15]. If there is enough time to send the data to at least one node with dirty data, sensors will transmit their information in the form of data streams containing their internal tables. Once sensors have exchanged their data, in the event of future encounters, the received request will be ignored. To begin the evaluation, sensors will only evaluate neighboring candidates who were within one hop during t_d.

3.3.1 Beta-Based Candidate Reduction

Our beta-based candidate reduction chooses a set of sensors who are the most correlated with respect to their trajectory behavior. We focus on the consideration of sensors within the transmission range at time t_d. Candidate sensors with the least similar behavior (i.e., the sensors with beta values below or above the lower and upper pre-defined boundaries L_b and U_b) are discarded. The ideal candidate sensor

would return a beta value of 1 for both betas, β_s and β_θ, meaning the sensor had the exact trajectory behavior as the sensor containing dirty data.

Trajectory Speed Beta

The similarity in trajectory speed variations along the sensors' trajectory path for a pair of nodes is measured by the speed beta, β_s, defined as:

$$\beta_s = \frac{cov(s_c, s_d)}{var(s_d)} \tag{3.1}$$

where
s_c: The speed values of the candidate sensor during T_e.
s_d: The are the speed values of the sensor with dirty data.

Angle of Travel Beta

The angle of travel represents the direction in which the sensor node is moving at the time instant the sensing is taking place. For a trajectory, the angle of travel beta, β_θ, is given by the equation:

$$\beta_\theta = \frac{cov(\theta_c, \theta_d)}{var(\theta_d)} \tag{3.2}$$

where
θ_c: The angle of travel values of the candidate sensor
θ_d The angle of travel values of the sensor with dirty data

3.3.2 m-Candidate Selection: Spatial Autocorrelation

The similarity in trajectory behavior evaluated above does not provide enough information in regards to the spatial correlation among the sensor nodes under analysis. Since the environmental data exhibits two main features: time stability and space correlation [KXL+13], we employ local Moran's I to identify spatial clusters most spatially autocorrelated during time t_d as depicted in Algorithm 1 Local Moran's I

identifies clustered sensor nodes with positive index values and outliers with negative index values. When evaluating the local Moran's I for the sensor containing dirty data, the missing value at time t_d is assigned the average of the sensed values collected in T_e. Local Moran's I is defined by:

$$I_i = \frac{\sum_{a=1}^{n} w_{ia}(z_a - \bar{z})(z_i - \bar{z})}{s^2 \sum_{a=1}^{n} w_{ia}} \tag{3.3}$$

where
$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$.
z: The z-score of the sensor evaluated at time t_d.
$w_{ia} = \alpha^{d_{ia}}$: An exponential function of the Euclidean distance d_{ia} between i and a.
α: A hyper-parameter specified using cross-validation.

After local Moran's I is calculated for all sensor nodes, the sensor containing dirty data selects m sensors with the smallest $|I_d - I_{c_i}|$ values as the set of most trustful candidate sensors. I_d and I_{c_i} are the local Moran's I of the sensor containing the dirty data and the candidate sensors, c_i, respectively.

3.3.3 Diversified Trust Portfolio Distribution

Based on the technique proposed in [Mar52], instead of devoting all our trust into one candidate sensor, we diversify the trust throughout a set of candidate sensors. Our DTP approach delivers a trust portfolio by assigning weights, w_i, to the different candidate sensors, c_i, based on the risk to trust each sensors' data. To find the weight that needs to be assign to each candidate sensor, it is necessary to calculate relative observation error:

$$E(w_1, w_2, ..., w_m) = \frac{\sqrt{\sum_{j \in \{t.\}_o^f \setminus t_d} \left(\frac{x_{dj} - \sum_{i=1}^{m} x_{cj} \times w_{ci}}{x_{dj}}\right)^2}}{|T_e| - 1} \tag{3.4}$$

where
m: The total number of candidate sensors

x_{dj}: The sensed value of the sensor containing dirty data time j

x_{cj}: The sensed value of the candidate sensor at time j

$|T_e| - 1$: The cardinality of the time instants selected for evaluation excluding t_d.

Since this is a continuous function and the domain of the function is compact, there is a minimum and a maximum. The diversified trust portfolio is generated by:

$$minE(w_1, w_2, ..., w_m) \tag{3.5}$$

such that $w_i \geq 0$ for any $i = 1, 2, ..., m$.

After the weights have been distributed among the candidate sensors to minimize the error, the estimated value, R, is computed as follows:

Algorithm 1: m-Candidate Sensor Selection

```
Input  : Time of missing value (t_d), and internal tables for sensor
         (d) containing dirty data and each candidate sensor at one
         hop (c_i)
Output: Set of m most trustworthy candidate sensors (C')
for c_i in C do
    β_s ← Equation (3.1)
    β_θ ← Equation (3.2)
    if L_b ≤ β_s ≤ U_b and L_b ≤ β_θ ≤ U_b then
        C' ← c_i          // Sensors that meet the trajectory behavior
                             requirements
    end
    I_d ← Equation (3.3)              // Local Moran's I of sensor d
    for c_l in C' do
        I_l ← Equation (3.3)
        δ ← {c_l, I_l}
    end
    C' ← m sensors with the smallest |I_d − I_{c_i}| values
end
return C'
```

$$R = \sum_{i=1}^{m} x_{c_i j} \times w_{c_i} \tag{3.6}$$

where

$x_{c_i j}$: The sensed value of the candidate sensor c_i at the dirty time j

w_{c_i}: The weight assigned by the trust portfolio to the candidate sensor c_i.

3.4 Simulations and Interpretation of Results

To evaluate the effectiveness of our approach, a number of environmental sensor nodes were placed in an area. The sensors select an initial sensor nodes' position, speed, and rest times, and continues to choose random destination points and speeds as per the steady state distributions of the random waypoint model outlined in [NC04]. Once the sensor arrives at the chosen point, it stops for a randomly selected time interval and continues to select another point and speed randomly.

We utilized Bonnmotion as our mobility generator together with MATLAB to simulate the environment and test our proposed approach. The simulation employs 50 minute time windows divided into two phases: sensing phase of 42.5 minutes and cleaning phase of 7.5 minutes. At the same time, each 50 minute time window is divided into 30 second time instants, in which sensors will begin to sense randomly. Our assessment is executed in indoor and outdoor environments.

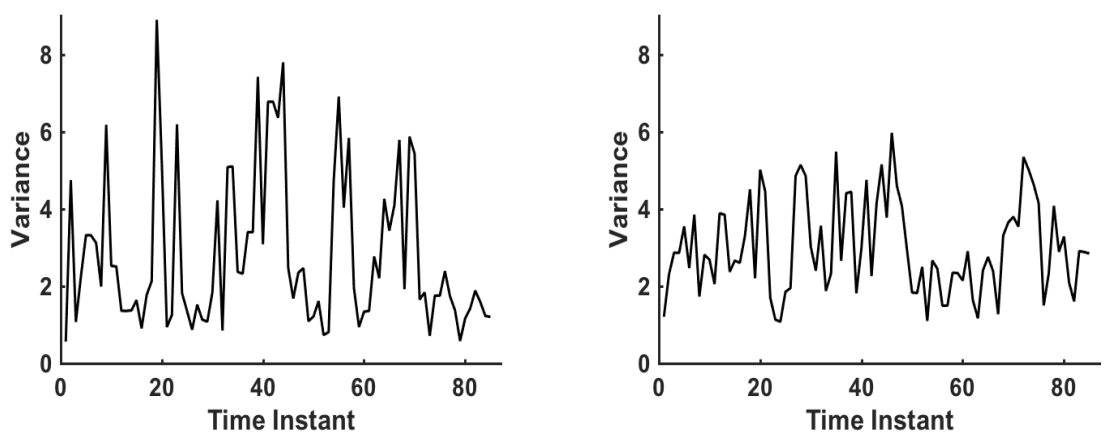
The values for L_b, U_b, α and m are -0.50, 2.00, 0.10 and 5 respectively, which are hyper-parameters for our selected dataset determined using cross-validation. The node densities tested are 1 and 2.5 nodes per 100 m^2. The number of sensors employed reaches up to 2,250 nodes with up to 95,625 dirty samples. The efficiency of the proposed technique was evaluated by calculating the average percent error of all cleaned samples at each time instant. Additionally, we calculated the cleaning level percentage for thresholds ranging from 0.50% to 10%. A sample is considered to be successfully cleaned if the absolute value of the percentage error between the value of reference and the calculated value falls below the specified error threshold. To contrast our results with existing techniques we selected Mean [JAF+06] and

LLSE [SGG10]. We specifically employed Mean's point, smooth and merge steps to detect sensor value outliers and correct missing data.

Intel Lab Data

During our indoor environment simulation we used the data provided by the Intel Indoor experiment [int], where 54 Mica2Dot static sensors collected various environmental data in an area of 1,200 m^2. This humidity data was employed to establish the values of reference for the area where our deployed mobile sensors collected data. The mean speed and transmission range used by each sensor were 2 miles per hour and 5 meters respectively.

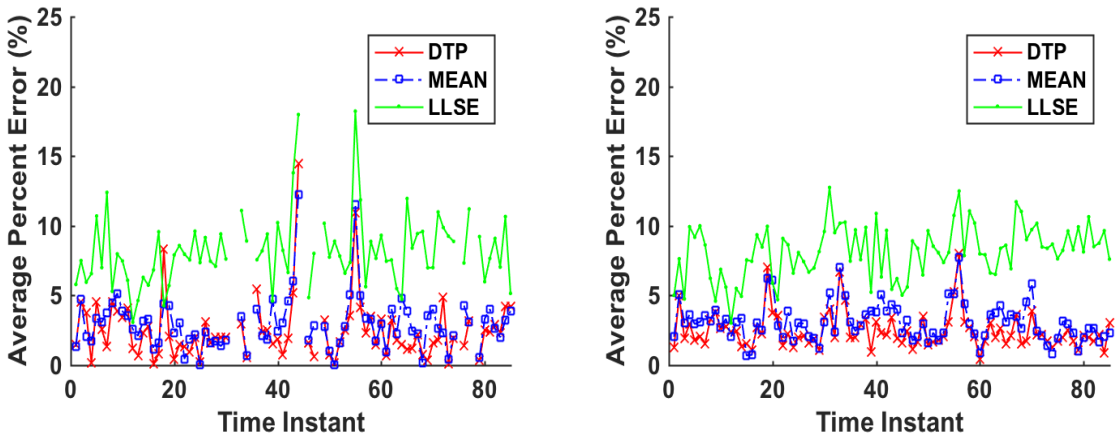
When the simulation was done in a low-density environment, the variance of the data collected by all sensors fluctuates drastically as shown in Figure 3.2(a). The discontinuities in the graphs occur when there were no values to be cleaned. Figure 3.3 shows that the average percent error of cleaned data by DTP stayed below Mean and LLSE when the simulation was performed with 20% and 50% of dirty data.



(a) 1 node per 100 m^2 (b) 2.5 node per 100 m^2

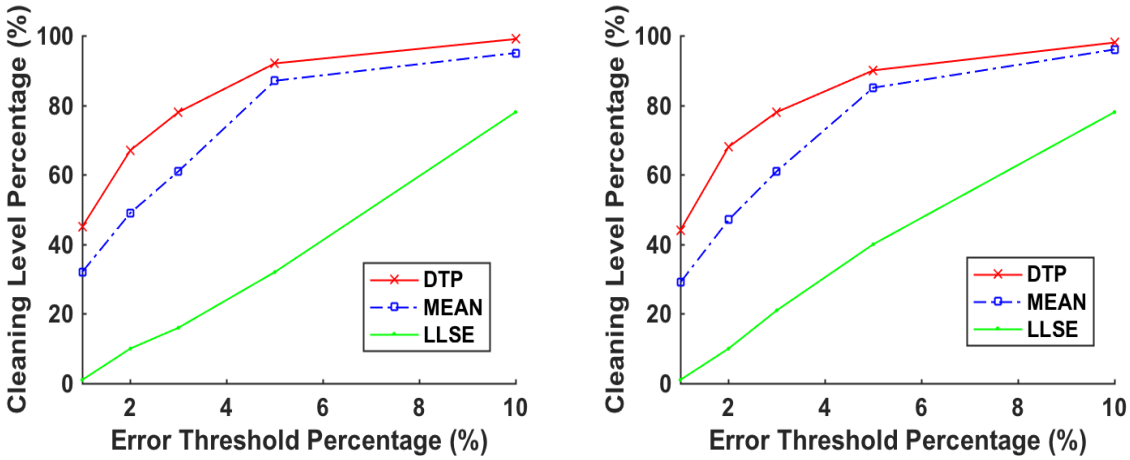
Figure 3.2: Collected Data Variance for Intel Lab Data.

Figure 3.4 confirms the performance of DTP as the cleaning level percentage reaches above the 98% under the 10% error threshold and above the 44% under 1% error threshold.



(a) 20% Dirty Data (b) 50% Dirty Data

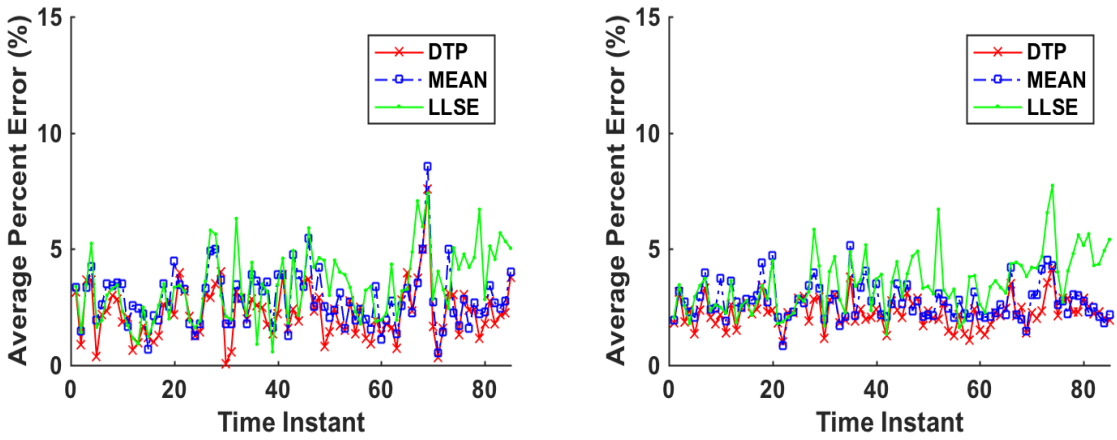
Figure 3.3: Data cleaning level in 1 node per 100 m^2 for Intel Lab Data.



(a) 20% Dirty Data (b) 50% Dirty Data

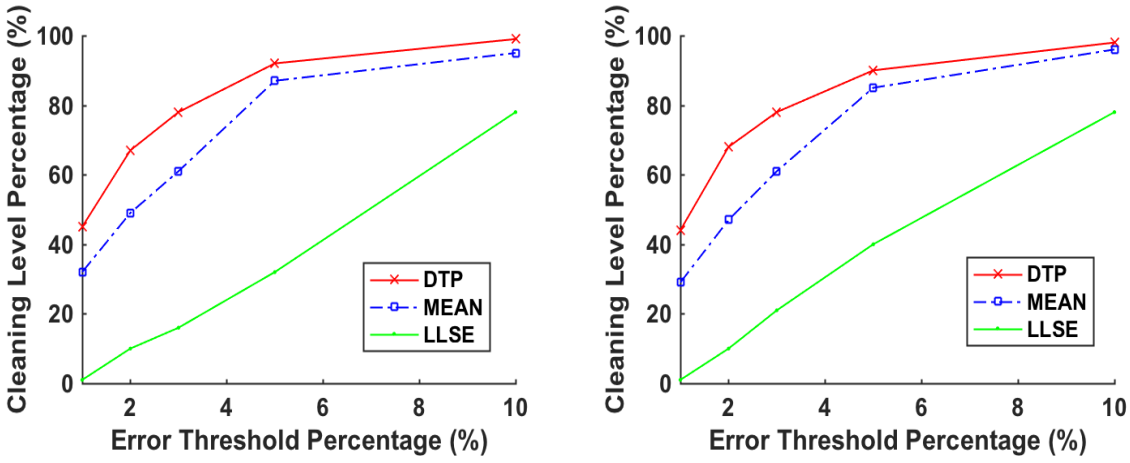
Figure 3.4: Cleaning level percentage at 1 node density for Intel Lab Data.

When the node density was increased to 2.5 nodes per 100 m^2, the variance of the data collected increased in stability, shown in Figure 3.2(b). Since LLSE takes into account the covariance among each sensors' data, its performance improved with the increase of the data stability, shown in Figure 3.5. Figure 3.6 displays our DTP approach kept a consistent performance over 99% of cleaning level percentage at 10% of error threshold and over 49% when tested for 1% error threshold.



(a) 20% Dirty Data (b) 50% Dirty Data

Figure 3.5: Data cleaning accuracy at 2.5 node per 100 m^2 for Intel Lab Data.



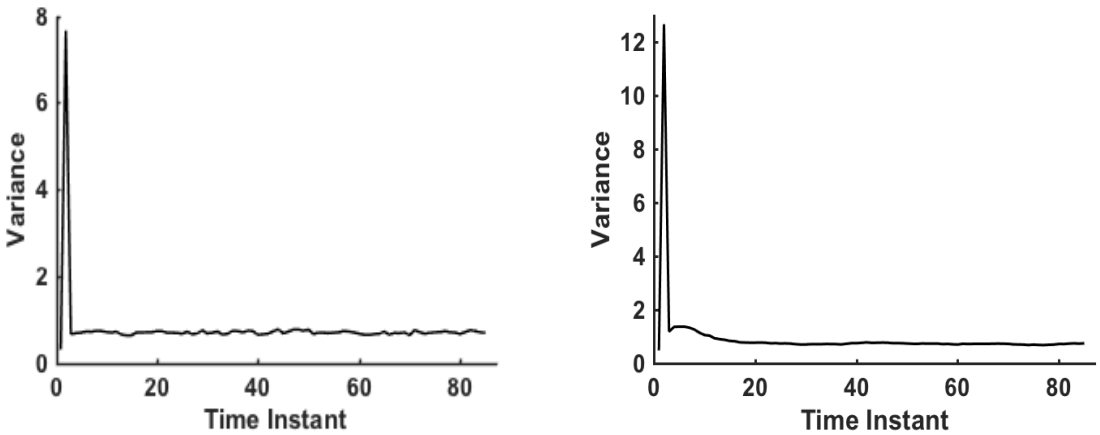
(a) 20% Dirty Data (b) 50% Dirty Data

Figure 3.6: Cleaning level percentage at 2.5 node density for Intel Lab Data.

Melbourne Weather Data

For our outdoor environment simulation, we used the Melbourne weather dataset [mel], where sensors collected temperature at 8 different locations from February 23-28, 2015. This collected data was used to generate the values of reference when evaluating the performance of our proposed method. In this assessment, sensors move in an area of 90,000 m^2 with a mean speed and transmission range of 20 miles

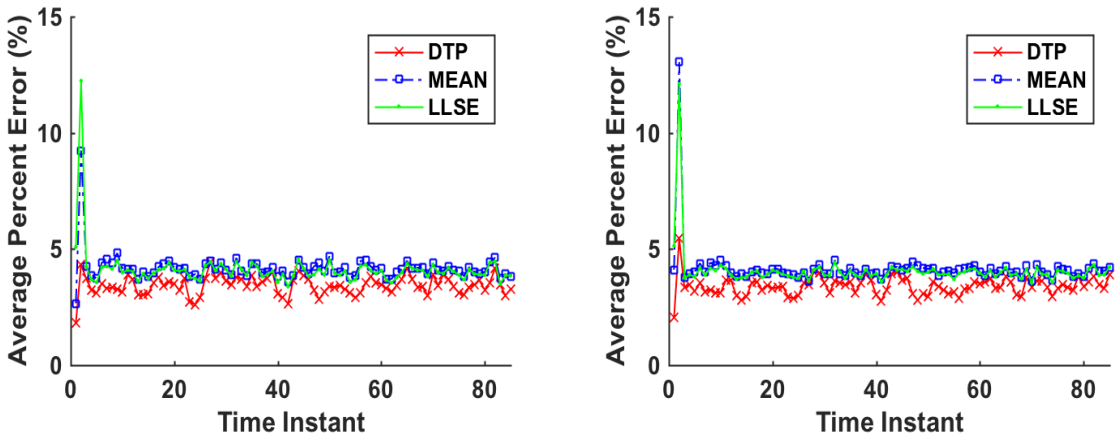
per hour and 15 meters respectively. From Figure 3.7 (a), it can be observed that the variance of the data collected by all sensors displays minimal fluctuations. The average percent error cleaned, shown in Figure 3.8, demonstrate DTP still maintains a lower average percentage error than Mean and LLSE methods.



(a) 1 node per 100 m^2 (b) 2.5 node per 100 m^2

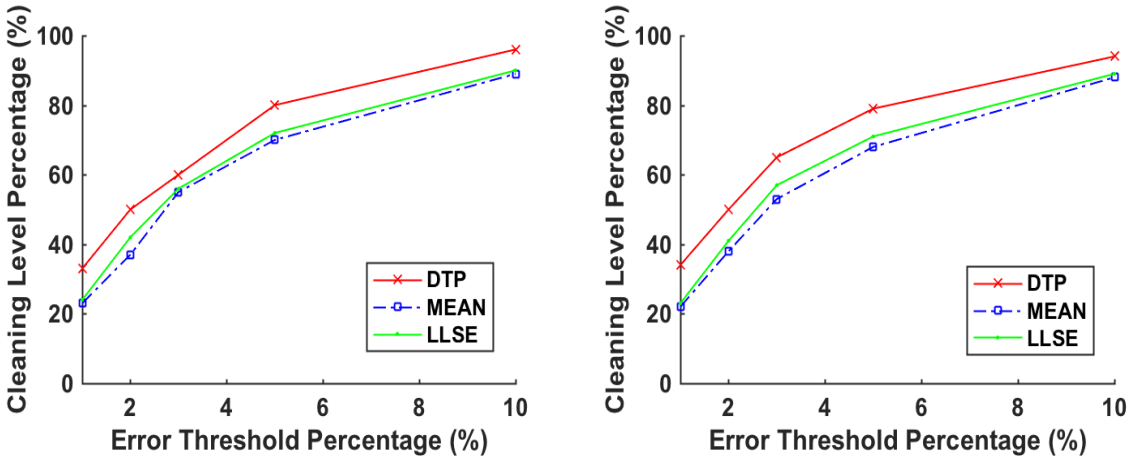
Figure 3.7: Collected Data Variance for Melbourne Data.

Figure 3.9 support the high efficiency of DTP by displaying over a 95% of cleaning level percentage for 10% error threshold and 49% for 1% error threshold. Similar to Figure 3.7(a), 3.7(b) shows a high variance on the data initially, while the performance of Mean and LLSE are affected by this variance, DTP keeps a consistently high performance as seen in Figure 3.10.



(a) 20% Dirty Data (b) 50% Dirty Data

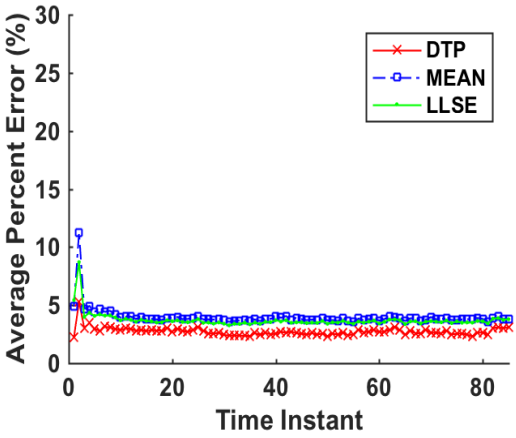
Figure 3.8: Data Cleaning accuracy at 1 node per 100 m^2 for Melbourne Data.



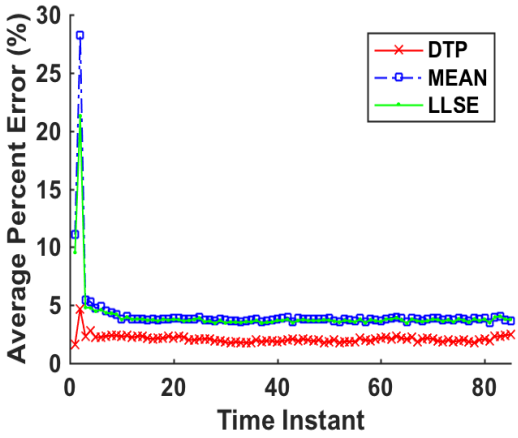
(a) 20% Dirty Data (b) 50% Dirty Data

Figure 3.9: Cleaning level percentage at 1 node density for Melbourne Data

Figure 3.11 shows that the cleaning level percentage was able to reach 98% under 1% error threshold and 59% for 10%. The consistent results of DTP are justified by its dependency on sensors' spatial autocorrelation and trajectory behavior rather than on the collected data only.

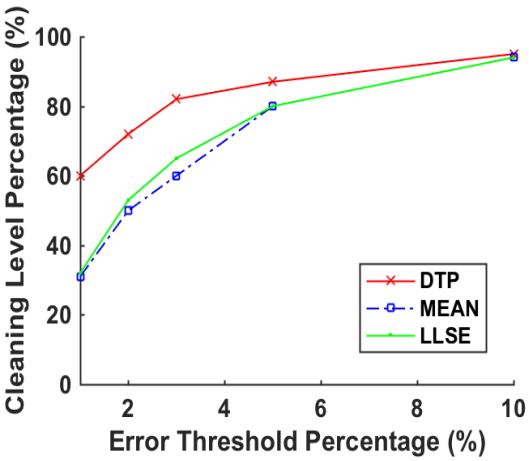


(a) 20% Dirty Data

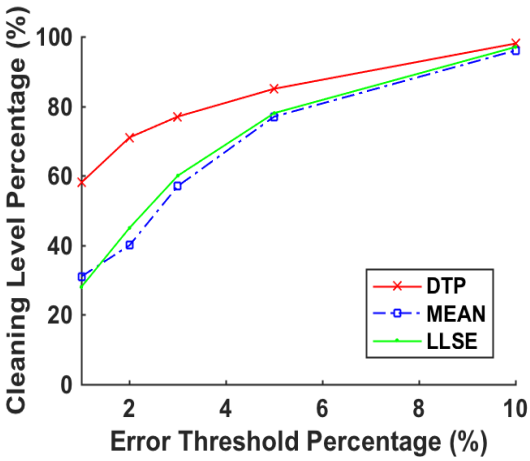


(b) 50% Dirty Data

Figure 3.10: Data Cleaning accuracy 2.5 node per 100 m^2 for Melbourne Data.



(a) 20% Dirty Data



(b) 50% Dirty Data

Figure 3.11: Cleaning level percentage at 2.5 node density for Melbourne Data

3.5 Summary

Our unique Diversified Trust Portfolio (DTP) approach for cleaning data in MWSN has demonstrated to be an effective method that utilizes the Spatio-temporal correlated data integrated with the analysis of each sensors' trajectory behavior to analyze the relationships among sensors. This constitutes an effective online system for selecting a trustworthy set of sensors to help during the in-network data cleaning process. By selecting a set of sensors, DTP is able to find the combination of weights to more accurately predict the values to clean the dirty data. This diversified trust technique reduces the risk involved in trusting a single sensor node's data. DTP demonstrated its outstanding capabilities to consistently achieve high data accuracy in comparison to two reputable data cleaning methods.

CHAPTER 4

A DYNAMIC TRUST WEIGHT ALLOCATION TECHNIQUE FOR DATA RECONSTRUCTION IN MWSN

Data accuracy and low energy consumption in MWSN are crucial attributes for real-time applications. Although there are many existing methods to reconstruct data for wireless sensor networks, there are few developed for highly mobile environments. In this chapter we propose a novel in-network data reconstruction method that determines the trust level in the data accuracy of each candidate node by evaluating Spatio-temporal correlations, trajectory behavior, quantity and quality of data, and the number of hops traveled by the received data from the source. Our proposed method is capable of evaluating second-hand data when there is no first-hand data available and selecting second-hand data the second-hand data is more accurate than the first-hand data. The evaluation of our results shows our method achieved lower and stable RMSE compared to other methods when predicting the missing data in scenarios with up to 70% of missing data. This chapter is organized as follows: Section 4.1 presents an introduction to data reconstruction in MWSN. Section 4.2 describes the problem statement. Section 4.3 describes the methodology. The proposed method is presented in Section 4.4. The evaluation of results and the summary can be reviewed in Section 4.5 and Section 4.6, respectively.

4.1 Introduction

Sensing data has gained increased importance in today's applications because it serves a means for understanding our surroundings. The accuracy of this data is a crucial component for real-time decision-making applications in MWSN. Nevertheless, sensor nodes in real-world MWSN data loss follow a set of different patterns that can result from noise, data collision, unreliable links or sensor nodes malfunction. Added to these challenges, the mobility of nodes reduces the chances of finding a set of sensor nodes that could help during the reconstruction of the data with high accuracy.

Moreover, when missing data in data streams reaches high levels, the energetic cost of reconstructing these data streams in an in-network fashion can be elevated.

It is known that in MWSN the maximum amount of energy is consumed by the communication process, which includes the transmission and reception of data. The second greatest energy consumer is computational operations in in-network processing. However, compared to the communication process, the computational operations expend much less energy [AQAKS17]. To mitigate the negative effects these limitations impose on mobile networks, the development of a light-weight method that seeks to reduce energy consumption while providing highly accurate data is necessary.

In the existing highly effective data reconstruction methods described in chapter 2 of this dissertation, nodes train models and upload them to the sink, and the utilization of this same model reconstructs sensed values. Nevertheless, in mobile environments, the delays produced by model training processes and the constant availability of a sink are unrealistic. Our proposed method is designed for real-time applications in mobile environments. Besides the trust in data accuracy concepts we employed in the previous method, we evaluate the quantity of high-quality data obtained from each node, and the nodes traveled to reach the evaluating point. This approach evaluation is based on the dynamic allocation of weights that indicates how trustworthy the data is without any threshold limitation.

4.2 Problem Statement and Assumption

It has been already discussed in this dissertation that the location of sensor nodes resulting from the mobility of nodes challenges the ability of sensor nodes to estimate the missing data accurately. As depicted in Figure 3.1 sensor nodes in MWSN applications can be embedded into vehicles or devices carried by people, and they can communicate to exchange information while moving in a determined pattern. Sensor nodes may have shared similar trajectory behavior and Spatio-temporal characteristics during the period in which either node experiments a loss of data. However, at a later time when a specific sensor node is carrying out its data reconstruction task, the nodes that were in its surroundings previously may have been far gone. Also, the sensors that are within communication range, may lack of the data re-

quired to reconstruct the data accurately. Is for this reason that in this method we consider the use second-hand data. We define second-hand data as any received data originated in a node other than the sensor node that transmitted it to the final destination.

It is assumed that each sensor node has an internal pre-process for the detection and elimination of dirty data, resulting in missing data. Data that is not detected as missing is considered valid and to be used for reconstruction. Our proposed method assumes a decentralized, in-network computational method. In other words, the selection of the candidate sensor(s) to perform the reconstruction, and the reconstruction itself, is done by each node. The sensing task takes place asynchronously, and the sensors are considered to be mobile, cooperative, and to have *a priori* knowledge of the area. The data exchange can take place between any two nodes via multi-hop communication and follow the Round Robin Scheduling Technique, as described in [SJG15].

4.3 Methodology

4.3.1 Data Loss Patterns

It is known that MWSN exhibit different types of data loss patterns [KXL+13]. The most commonly employed data loss patterns in real-time applications are:

- **Element random loss pattern:** Elements are dropped independently and randomly during the transmission. This can be as a result of noise and collision.
- **Element frequent loss in a row pattern:** Sensed data from a single node has higher probabilities of loss. This can be produced by unreliable links.
- **Successive element loss in a row:** Sensor stops sensing at some point in time and produces no more sensed values until the end of the simulation. This could be a result of sensors losing energy or malfunctioning.

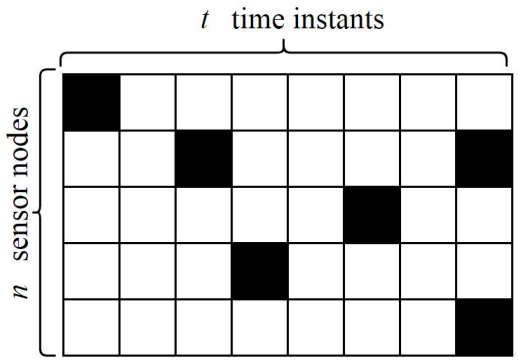


Figure 4.1: Element Random Loss

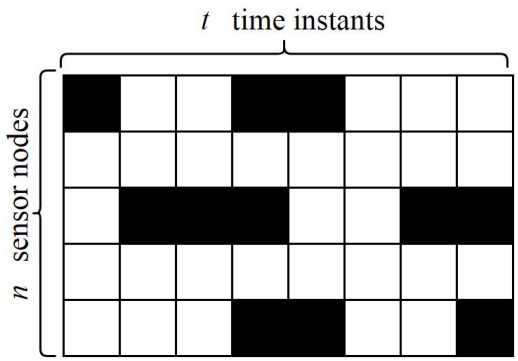


Figure 4.2: Element Frequent Loss

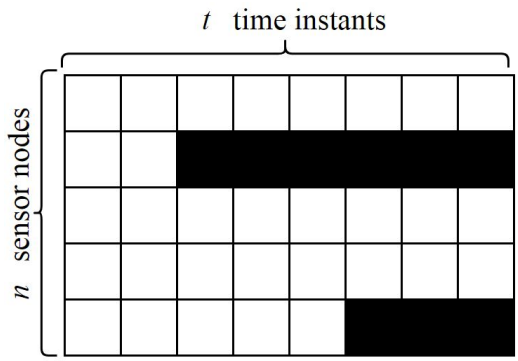


Figure 4.3: Successive Element Loss in a Row

DTWA is designed to reconstruct real-life data loss patterns which include a combination of all the aforementioned patterns. Figures 4.1, 4.2 and 4.3 give a graphic view of the described data loss patterns in MWSN.

4.3.2 Data Gathering

In our DTWA approach, data is gathered and reconstructed in a time window T, which is partitioned into (1) sensing phase T_s and (2) reconstruction phase T_r; at the same time T is divided into t time instants. Each time instant is appointed as t_j, where $j = 1, 2, ..., t$.

4.3.3 Sensing Phase (T_s)

Each node will sense and collect data. At the end of every t_j, sensors will update their internal tables, as shown in Table 4.1, using the sensed value (x_{ij}), its position (l_{ij}), angle of travel (θ_{ij}), and average speed (s_{ij}), where $i = 1, 2, ..., n$, and n is the total number of sensors in the network. When the node detects a dirty sample within its sensed values, the value is marked with a flag as missing. When a sensor is reconstructing its data, it will select the sub-partition of time, T_e (evaluation time), within T_s to carry out the evaluation of candidate sensors. The sub-partition of time can be described as: $T_e = t_m - k, ..., t_m - 2, t_m - 1, t_m, t_m + 1, t_m + 2, ..., t_m + p$, where t_m is the time instant where there is a missing value. The initial and final time instants in T_e are $t_o = t_m - k$ and $t_f = t_m + p$, respectively. $|T_e|$ is user-defined and, k and p are dynamic values based on the position of t_m.

Time t_j	1	2	...	j
Location l_{ij}	l_{i1}	l_{i2}	...	l_{ij}
Sensed Value x_{ij}	x_{i1}	x_{i2}	...	x_{ij}
Average Speed s_{ij}	s_{i1}	s_{i2}	...	s_{ij}
Angle of Travel θ_{ij}	θ_{i1}	θ_{i2}	...	θ_{ij}
Missing Data Flag f_{ij}	f_{i1}	f_{i2}	...	f_{ij}

Table 4.1: Sensor i Internal Table for T_s

Reconstruction Phase (T_r)

In this period, if sensors have data to be reconstructed, at every time t_j, sensors will broadcast a data request message containing its identification and current location. Receiving sensors estimate if there is enough time to send their data to the requesting sensor before the two move out of the transmission range. Once sensors confirm the feasibility of the transmission, sensors will transmit their information together with the information from any other node received during T_r.

4.4 Dynamic Trust Weight Allocation (DTWA)

Our Dynamic Trust Weight Allocation (DTWA) method quantifies the level of trust in data accuracy for each of the candidate sensors without the usage of predefined thresholds. The DTWA scheme revolves around common factors and conditions faced by each node. When a sensor node contains a missing value, it must evaluate influencing factors to identify which node(s) data accuracy to trust. DTWA describes trustworthy nodes as nodes containing the highest quantity of high-quality, spatiotemporally correlated data with a significant resemblance in trajectory behavior about the evaluating node. To compute the total trust (τ), we evaluate the following parameters:

- **Confidence level (ϕ_c):** To evaluate the trustworthiness of the accuracy of the data provided by a node, taking into consideration the number of correct observations provided.
- **Spatio-Temporal Closeness (ϕ_x):** To prioritize first-hand data yet, consider second-hand data while taking into account the Euclidean distance between two nodes.
- **Pearson's Correlation Coefficient (ρ):** To quantify the strength of a linear relationship between the collected sensed values for the pair of sensors.
- **Normalized Speed Beta (β_s):** To quantify the similarity in trajectory speed variations for a pair of sensors.

- **Normalized Angle of Travel Beta (β_θ):** To measure the similarity in direction of a pair of sensors. The angle is calculated at the time instant the sensing is occurring, and are values from a point of interest relative to a given axis.

After all the parameters have been evaluated, the total trust in the data accuracy of an individual node can be computed. Figure 4.4 shows an example of an MWSN, in this scenario, all sensor nodes may have provided their sensed data and depicted a strong correlation among their sensed values. Although nodes may receive the data of each candidate node via one-hop communication and share similar locations/trajectories at time t_1, the change of speed after time t_1 can be the determining factor if a node is attempting the reconstruction of data after time t_1.

Trust Parameters Evaluation

The degree of trust evaluation is an adaptive mechanism to assess the certainty in data accuracy. Each evaluating node performs this evaluation for each candidate sensor with the data provided. To carry out this evaluation, we consider the following parameters:

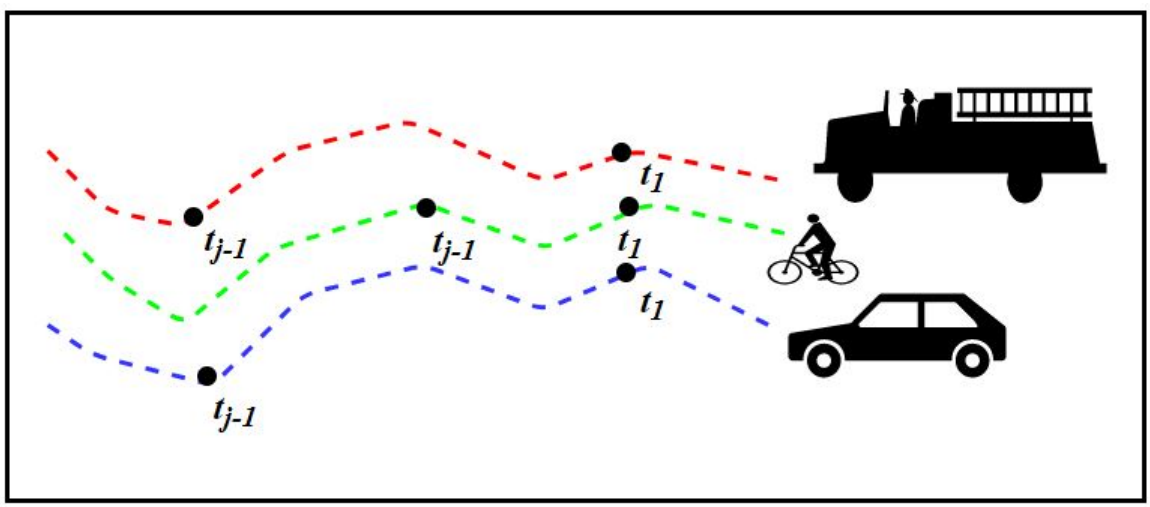


Figure 4.4: Trajectory Behavior Example

a) **Confidence Level**

The confidence level quantifies the number of samples provided versus the number of missing samples during the evaluation time T_e. This value ranges from [0,1], where 0 indicates that the node did not provide any valid sensed sample and 1 represents that the node provided all valid samples and no missing samples. The confidence level can be calculated using a modified formula from [SDBA15], and is given by:

$$\phi_c = 1 - \sqrt{\frac{12 \times \delta \times \xi}{(\delta + \xi)^2(\delta + \xi + 1)}} \tag{4.1}$$

where

δ: The number of time instances containing complete, accurate data.

ξ: The number of time instances with missing data for the candidate sensor, y.

b) **Spatio-Temporal Closeness**

The Spatio-temporal closeness parameter quantifies how close two sensor nodes were in t_m. It considers not only the distance among the two nodes but whether the data was received via one-hop or multi-hop. In other words, if the data received is first-hand or second-hand. The Spatio-temporal can be determined as shown below:

$$\phi_x = \begin{cases} (\alpha)^d & \text{if received via one-hop} \\ (\gamma)^d & \text{if received via multi-hop} \end{cases} \tag{4.2}$$

subject to $\alpha > \gamma$

where

α: Hyper-parameter assigned when data was received via one-hop.

γ: Hyper-parameter assigned when data was received via multi-hop.

d: The Euclidean distance between the sensors at the time to be reconstructed.

c) **Pearson's Correlation Coefficient**

Pearson's correlation coefficient [Pea95] calculates the correlation between the sensed values during the evaluation period T_e. For a pair of nodes, a positive correlation indicates sensed values are directly proportional. A negative correlation indicates sensed values are inversely proportional. The closer Pearson's correlation coefficient approaches 1, the stronger the positive linear relationship. DTWA considers only positive Pearson's correlation coefficients ranging from [0,1] and negative coefficients ranging from [-1,0) are assigned 0 at evaluation time. Namely, sensors with inverse relationships are given no trust. Pearson's Correlation coefficient is given by:

$$\rho = \frac{cov(x, y)}{\sigma_x \times \sigma_y} \tag{4.3}$$

where

x: The sensed values during T_e of the sensor with data to be reconstructed.

y: The sensed values of the candidate sensor during T_e.

σ_x: The standard deviations for x.

σ_y: The standard deviations and y.

d) **Trajectory Behavior Similarity**

To compare the trajectory behavior between the baseline node and each candidate node, we utilize Beta analysis [Dam, SAPA$^+$18]. Specifically, we compute Speed Beta (β_s), and Angle of Travel Beta (β_θ). Betas quantify how similar or dissimilar each node behaves in relation to the evaluating node. The ideal candidate sensor would return a beta value of 1 for both betas, β_s and β_θ, meaning that both sensors had the exact trajectory behavior. The similarity in trajectory variations along the sensors' trajectory path for a pair of sensors is computed as:

$$\beta_u = \frac{cov(u_c, u_m)}{var(u_m)} \tag{4.4}$$

where u_c and u_m: are the speed or angle of travel values of the candidate sensor during T_e and the sensor with missing data, respectively. The beta values are

normalized to be transformed into weight coefficients. The new normalized values represent the probability that the value could appear in the given historical data. To obtain the normalized beta values, β'_u, we compute the following formula:

$$\beta'_u = \frac{\beta_u - min(\beta_u)}{max(\beta_u) - min(\beta_u)} \tag{4.5}$$

where

β_u: The non-normalized beta values.

$min(\beta_u)$: The minimum values of all the collected beta values that a sensor contains.

$max(\beta_u)$: The maximum values of all the collected beta values that a sensor has.

Total Trust Computation

Combining the five parameters described above results in our total trust formula. The total trust is computed as follows:

$$\tau = \frac{(w_1\phi_c) + (w_2\phi_x) + (w_3\rho) + (w_4\beta'_s) + (w_5\beta'_\theta)}{w_1 + w_2 + w_3 + w_4 + w_5} \tag{4.6}$$

where

ϕ_c: The confident level.

ϕ_x: The spatio-temporal closeness.

ρ: The Pearson's Correlation coefficient.

β'_s: The normalized speed beta.

β'_θ: The normalized angle of travel beta.

w_1, w_2, w_3, w_4 and w_5: User-defined weights assigned to each parameter.

Once the sensor containing missing data has computed the total trust (τ) for each candidate sensor, it will select the sensor(s) with the highest value (τ), as shown in Algorithm 2. Finally, the selected sensors' information is used to approximate the missing data.

Algorithm 2: Candidate Sensor(s) Selection

Input : Internal tables for the sensor (d) and each candidate sensor (c_i), time of missing value (t_m), and user-defined potential number of candidate sensors (n_c)

Output: Set of most trustworthy candidate sensors

```
for c_i in C do
    φ_c(c_i) ← Equation (4.1)                          // Confidence Level
    if φ_c(c_i) = 1 then
        C_complete ← c_i          // Set of sensors with complete data in T_e
    end
    φ_x(c_i) ← Equation (4.2)                 // Spatio-Temporal Closeness
    ρ(c_i) ← Equation (4.3)                     // Sensed Value Correlation
    β_s(c_i) ← Equation (4.4)                                  // Speed Beta
    β'_s(c_i) ← Equation (4.5)                    // Normalized Speed Beta
    β_θ(c_i) ← Equation (4.4)                        // Angle of Travel Beta
    β'_θ(c_i) ← Equation (4.5)          // Normalized Angle of Travel Beta
    τ(c_i) ← Equation (4.6)                              // Total Trust Value
    if |C_complete| ≥ n_c then
        C' ← Set of sensors with the greatest τ values from C_complete
    end
    else
        C' ← Single sensor with the greatest τ value in C
    end
end
return C'
```

Missing Value Approximation

If at least n_c candidate nodes with $\phi_c = 1$ exist, the diversified trust portfolio and data predictions are made using Equations (4.7) and (4.8). If that does not exist, then linear regression is employed is employed. Equations (4.9) and (4.10) are employed for the prediction. Where n_c is a user-defined parameter that specifies the minimum quantity of candidate sensors are preferred to employ the diversified trust portfolio technique.

a) $\exists\ n_c$ **Candidate Sensors with** $\phi_c = 1$

Based in the portfolio selection technique proposed in [Mar52], instead selecting one single candidate node, we diversify the trust throughout a set of candidate sensors. Choosing a set of candidate sensors can help to minimize the error between the predicted data and the real values. The diversified trust portfolio assigns weights to multiple candidate sensors based on the risk to trust each sensors' data [SAPA+18]. To find the weight to be assigned to each candidate sensor, it is necessary to minimize the relative observation error:

$$E(w_1, w_2, ..., w_{n_c}) = \frac{\sqrt{\sum_{j \in \{t.\}_o^f \setminus t_m} \left(\frac{x_{dj} - \sum_{i=1}^{n_c} x_{cj} \times w_{ci}}{x_{dj}}\right)^2}}{|T_e| - 1} \tag{4.7}$$

where

n_c: The total number of candidate sensors.

x_{dj}: The sensed values of the sensor missing data at time j.

x_{cj}: The sensed values of the candidate sensor during time j.

w_{ci}: The weight variable to be minimized.

$|T_e| - 1$: The cardinality of the set of time instants under evaluation, excluding t_m.

t_m: The time containing the missing data we are approximating.

This is a continuous function with a compact domain, so there is a guaranteed minimum and maximum. To find the combination of weights that will depict the highest precision when approximating the missing value, the diversified trust portfolio is generated by minimizing $E(w_1, w_2, ..., w_{n_c})$ such that $w_i \geq 0$ for any $i = 1, 2, ..., n_c$. After the weights have been spread out among the candidate sensors to minimize the error, the estimated value, R, is calculated as follows:

$$R = \sum_{i=1}^{n_c} x_{c_i m} \times w_{c_i} \tag{4.8}$$

where

$x_{c_i m}$: The sensed value of the candidate sensor c_i at the missing time m.

w_{c_i}: The weight assigned by the trust portfolio to the candidate sensor c_i.

b) $\nexists\ n_c$ **Candidate Sensors with** $\phi_c = 1$

If there are no sensors that do not contain an missing values in T_e, then we select one candidate sensor with the highest trust value τ. To approximate the value at the time with missing data, we calculate the r-correlation coefficient [MPV12] as below:

$$r = \frac{\sum_{i=t_o}^{t_f} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=t_o}^{t_f} (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{4.9}$$

where $b_1 = r\frac{\sigma_x}{\sigma_y}$.
$b_0 = \bar{y} - (b_1 \times \bar{x})$.
x: The values for the sensor containing missing data.
y: The sensed values for the sensor the candidate sensor.
$\bar{x}$: The mean of the sensed values of the sensor containing missing data.
$\bar{y}$: The mean of the sensed values the candidate sensor.
y_m: The sensed value of the selected candidate sensor at t_m.

The missing value, R, is approximated as:

$$R = b_0 + (b_1 \times y_m) \tag{4.10}$$

where
b_0: The estimate of the regression intercept.
b_1: The estimate of the regression slope.
y_m: The value provided by the candidate sensor that is employed for the prediction.

4.5 Simulations and Interpretation of Results

To evaluate the performance of our method, a number of mobile sensor nodes were placed in an area. The sensors followed the steady state random waypoint mobility model [NC04]. We utilized the Bonnmotion Mobility Scenario Generation and Analysis Tool to generate sensor mobility and MATLAB to simulate our testing environment and perform the data cleaning computations. In this assessment, sensors move with a mean speed and transmission range of 20 miles per hour and 15 meters, respectively. For our simulation, we used the Melbourne dataset [mel], where sensors collected temperature at 8 different locations from February 23-28, 2015. This collected data was used to generate the values of reference when evaluating the performance of our proposed method. We employed a 50-minute time window divided into two phases: sensing phase of 42.5 minutes and reconstruction phase of 7.5 minutes. Every time instant t_j had a duration of 30 seconds and sensing occurred asynchronously during every time instant of the sensing phase.

The values for α and γ are 1 and 0.01, respectively, and are the hyper-parameters for our selected dataset evaluated using cross-validation. The weight parameters w_1, w_2, w_3 w_4 and w_5 employed to calculate total trust are equal to 1, as we consider all parameters to be equally important. $|T_e|$ and n_c are 10 and 5, respectively. The simulation was performed with 900, 450 and 250 nodes in an area of 90,000 m^2. The percentage of missing samples were 20%, 50%, and 70%. The performance of our technique was evaluated by calculating the Root Mean Square Error (RMSE) of all reconstructed samples at each time instant. RMSE is a quadratic scoring rule that measures the average magnitude of the error and is beneficial in penalizing large errors. To evaluate our results, we compared DTWA with IMC [ZSS14].

Table 4.2 shows the average percentages of missing data reconstructed using first-hand and second-hand data throughout our simulation. It also presents the average distribution of missing data generated by each of the data loss patterns simulated. As element frequent loss in a row is the most common data loss pattern in MWSN, up to 66.51% of the missing data was lost using that data loss pattern. Up to 22.51% and 20.54% of the missing data was generated using the element random loss pattern and the successive element loss in a row pattern, respectively.

Sensor Count n	**250**	**450**	**900**
First-Hand	42.07%	31.42%	36.71%
Second-Hand	57.93%	68.58%	63.29%
Element Random Loss	21.64%	22.51%	22.09%
Element Frequent Loss in a Row	58.20%	56.95%	66.51%
Successive Element Loss in a Row	20.16%	20.54%	11.40%

Table 4.2: Average Data Loss and Collection Statistics

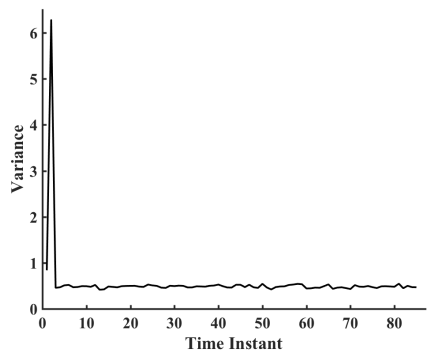
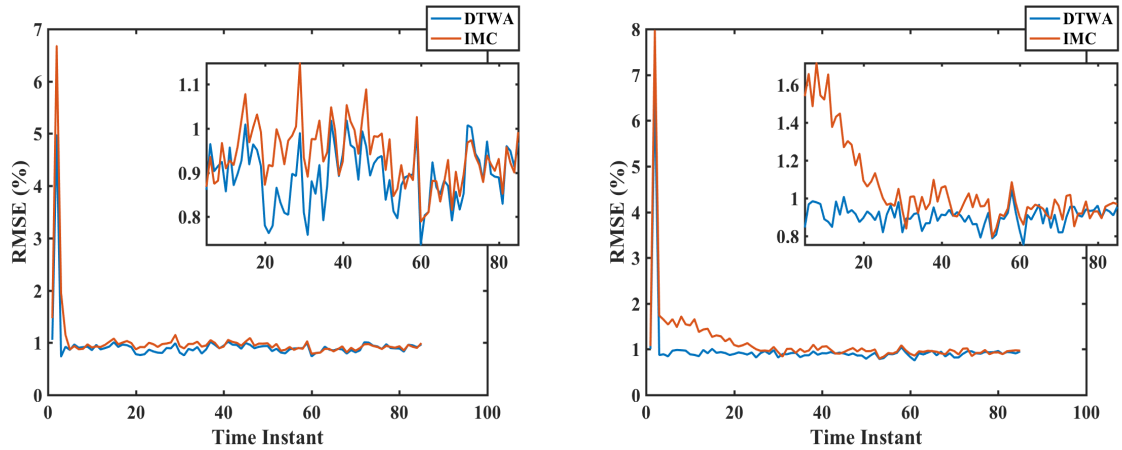


Figure 4.5: Collected Data Variance

Figure 4.5 shows a sample of the variance of the data collected by the 450 sensors' simulation. The variance of the data collected in all of our simulations had similar behavior with a spiked variance in the initial few time instances and a constant lower variance once all sensors have collected a few values. This is due to the steady state behavior of our sensors in our simulation. From the behavior of the data collected and the RMSE at each simulation, it is inferred that the RMSE and the variance are directly proportional.

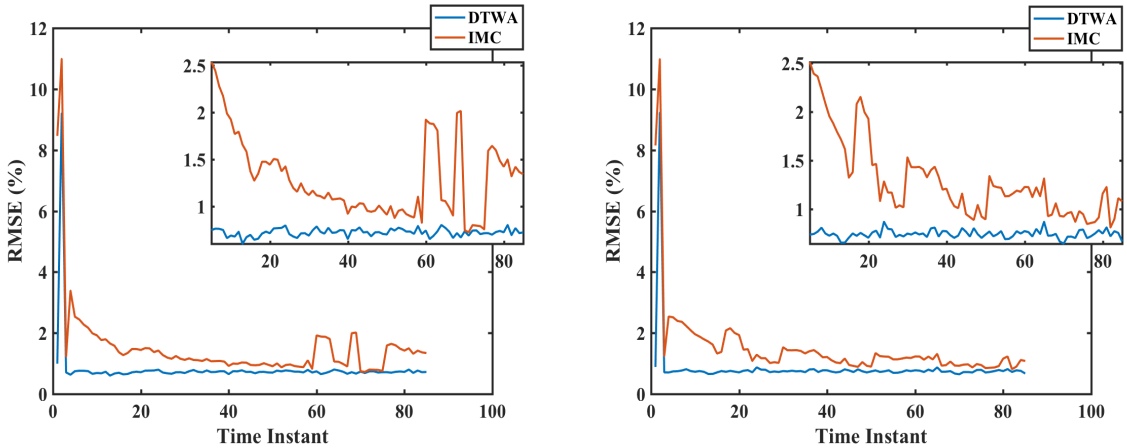
The RMSE of the values reconstructed by DTWA contrasted against IMC for 250 sensors' simulation is shown in Figure 4.6. When tested for 50% of incomplete data, IMC shows to be highly competitive compared to DTWA. However, when tested for 70%, it is evident that it is much harder for IMC to learn which sensor

is most trustworthy, while DTWA is resilient to high quantities of missing data. This is because IMC only considers Spatio-temporal and sensed value correlations while DTWA considers additional parameters that influence the data reconstruction process in real-life applications. Figures 4.6, 4.7, and 4.8 contain insets to more clearly see the RMSE at each time instant after the initial spike in the variance.



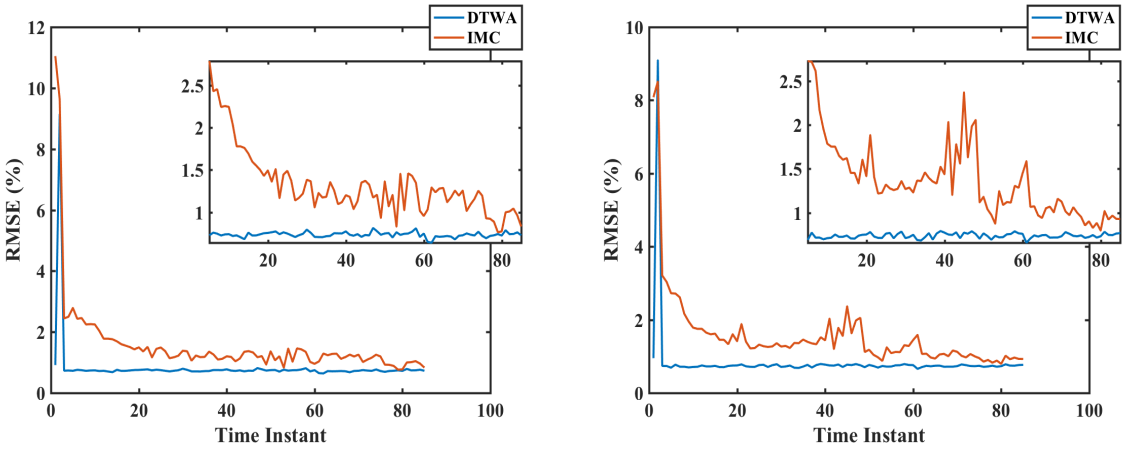
(a) 50% Dirty Data (b) 70% Dirty Data

Figure 4.6: 250 sensors collecting Melbourne Data.



(a) 50% Dirty Data (b) 70% Dirty Data

Figure 4.7: 450 sensors collecting Melbourne Data.



(a) 50% Dirty Data

(b) 70% Dirty Data

Figure 4.8: 900 sensors collecting Melbourne Data.

The likelihood of finding an optimal sensor with the most valid data and similar trajectory behaviors increases as the number of sensors increases. Utilizing the trust parameters evaluation, DTWA selected second-hand data in up to 68% of its predictions, as shown in Table 4.2. IMC shows spikes in their RMSE throughout the simulations because, at particular time instances, the amount of available valid data is minimal. DTWA depicts a low and stable RMSE regardless of how many missing values there were at a specific point in time.

The increase in the number of sensors complicates the selection of trustworthy sensor(s) in data reconstruction techniques. Figures 4.7 and 4.8 demonstrate that DTWA performed better compared to IMC by predicting the missing data with higher precision, even when the number of sensors and their interactions were increased. It was easier for DTWA to approximate the values because the prediction of the missing data depends on the evaluation of different behavior parameters in both first-hand and second-hand data.

4.6 Summary

DTWA is a novel and effective light-weight in-network technique designed to reconstruct highly incomplete datasets in mobile environments. Our scheme quantifies the level of trust in data accuracy for each candidate sensor and revolves around common factors and conditions faced by each node in real-world applications. DTWA's accuracy is obtained from the selection of sensor(s) with the highest quantity of high-quality, spatiotemporally correlated data and with a significant resemblance in trajectory behavior. DTWA can be easily tailored to different scenarios in MWSN, and the flexibility of the modification of weights given to each attribute can contribute to meet specific user requirements in diverse scenarios. The dynamic adaptive features of DTWA makes it suitable for evaluating the certainty of data accuracy for neighboring sensor nodes in scenarios with large quantities of missing data and sensor count, such as in IoT.

When compared to IMC, another useful light-weight algorithm, DTWA demonstrated its outstanding capabilities to consistently achieve high data accuracy with vast quantities of missing data. Since IMC showed to outperform LLSE and the Mean methods [SGG10, JAF+06], and DTWA outperformed IMC, we can derive that DTWA can achieve better data accuracy than the two well-known methods, LLSE, and the Mean. Contrary to various current methods, the evaluation of trust in DTWA is not affected by past interactions, which addresses the newcomer problem. DTWA is also an energy-aware method, as sensors will only compute predictions when there is missing data.

CHAPTER 5

USING CANDLESTICK CHARTING AND DYNAMIC TIME WARPING FOR DATA BEHAVIOR MODELING AND TREND PREDICTION FOR MWSN IN IOT

There is a rapid emergence of new applications involving MWSN in the field of the Internet of Things. Although useful, MWSN still carry the restrictions of having limited memory, energy, and computational capacity. At the same time, the amount of data collected in the Internet of Things is exponentially increasing. In this chapter we propose a data abstraction and trend prediction technique, called Behavior-Based Trend Prediction (BBTP), to address the limited memory constraint in addition to providing future trend predictions. Predictions made by BBTP can be employed by real-time decision-making applications and data monitoring. BBTP applies the Japanese Candlestick charting technique, popularly employed in financial markets to abstract the data behavior of a time partition in evolving data streams. It also quantifies differences between a pair of consecutive time partitions utilizing dynamic time warping at the sensor node. Then, it forwards the data to an Internet-enabled device, where the sensor's future data trends are predicted. Our results demonstrate that data trends predicted by BBTP achieve better precision, recall, and accuracy score when contrasted against four well-known techniques while reducing the space complexity by at least a factor of 10. This chapter is organized as follows: An introduction to predictive methods in MWSN is provided in Section 5.1. Section 5.2 presents the problem statement. Section 5.3 introduces the proposed method. The evaluation of results and the summary can be reviewed in Section 5.4 and Section 5.5, respectively.

5.1 Introduction

Mobile wireless sensor networks (MWSN) are essential elements of the Internet of Things (IoT) as they increase the coverage of the Internet and the expansion of computing [YH18]. The well-known resource constraints in these types of networks include limited memory, low computational capacity, and restricted power sources. It is for this reason data abstraction techniques that reduce the required stored and transmitted data and effectively model the evolving data streams are crucial. Although real-time monitoring and prediction of future data values are beneficial for decision support, the projection of data streams behavior trends is particularly relevant for applications that seek to take preventive actions. While the prediction of specific values can assist in taking preventative measures, the ability to foresee the direction of the data evolution may have the same impact without the added computation involved in the prediction of data values.

This chapter proposes a Behavior-Based Trend Prediction (BBTP) method that abstracts the behavior of data and predicts their future trends. BBTP consists of the use of three main algorithms: a Japanese candlestick charting technique for data abstraction, a similarity measure using dynamic time warping (DTW), and a multi-class Support Vector Machine (SVM) for trend prediction. The Japanese candlestick charting technique models the historical behavior of evolving data streams for a sensor node. Then, dynamic time warping (DTW) measures the similarity between consecutive time partitions to characterize the changes and progression of the extracted data over time. Lastly, the SVM learns the data behavior progression from the similarity measures obtained during the DTW stage and predicts future data trends. The data reduction propelled by BBTP results in multiple benefits, including network traffic reduction and energy preservation at the sensor node level, and a prolonged functionality of the network [BH14]. At the Internet-enabled device level, less data is required to train the model, without having to trade off the accuracy of predicted trends. The applicability of our proposed solution may extend to a variety of applications in IoT in which data streams possess an evolving behavior.

5.2 Problem Statement

The goal of this work is to reduce the amount of required stored data in sensor nodes. This reduction in data through data abstraction can lessen the network traffic and reduce the energy employed during the transmission of large volumes of data streams. The individual sensor nodes also serve to perform the data abstraction to aid in modeling the evolving behavior of data streams in an in-network fashion. An effective abstraction and modeling will result in accurate future data trend predictions. We consider a scenario as shown in Figure 5.1, where mobile sensors communicate to Internet-enabled devices. In this scenario, the data abstraction is performed by the sensor node and then data is forwarded to the IoT device, where trend predictions are computed. The predictions are sent to a server, where it can be stored and accessed through a dashboard for visual analysis and monitoring or which can be employed for decision-making in real-time applications.

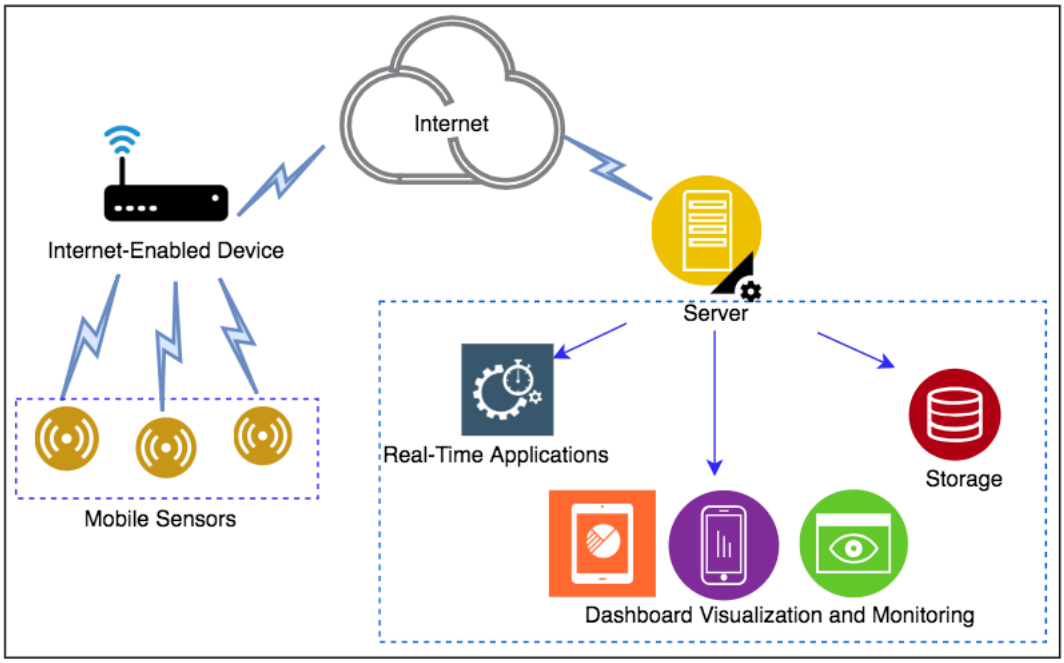


Figure 5.1: Mobile Wireless Sensor Networks in the Internet of Things

5.3 Behavior-Based Data Prediction

Our Behavior-Based Trend Prediction (BBTP) method abstracts the behavior of sensor's data generated in the form of data streams. For this behavior abstraction, BBTP employs a candlestick charting technique, commonly used in currency markets analysis. Once the data is partitioned, each time partition or candlestick contains the data of a user-defined time duration. In other words, each candlestick represents the data during this given time duration. Once the data is extracted, our method employs the Dynamic Time Warping (DTW) algorithm to quantify the behavior similarity between each consecutive pair of time partitions, or candlesticks. The trend prediction in our technique utilizes a multi-class SVM that learns the evolution of the behavior of the abstracted data to make the predictions. These three main steps are depicted in Algorithm 3, where the inputs are the sensor's evolutionary data streams ($\{x_1, x_2, x_3, ...x_n\}$) and the user-defined time duration (d_t).

In our analysis of data behavior, a sensor only evaluates the data it collects and not data collected by surrounding sensors. It mainly focuses on the progression of the sensor's historical information and its data evolution throughout a given period. In other words, our data behavior analysis is only concerned with the data a sensor generates about itself and its data evolution over time. Data behavior analysis assumes that the data evolution reflects the variables and factors that influence the sensing data and cause its behavior to change. Due to this reason, analyzing a sensor's historical data is enough to make predictions about future data trends

Algorithm 3: Behavior-Based Trend Prediction

Input : Sensor data steams ($\{x_1, x_2, x_3, ..., x_n\}$), user-defined time duration (d_t)

Output: Data stream of predictions (*pred*)

1 $S \leftarrow$ CANDLESTICKS($\{x_1, x_2, x_3, ..., x_n\}$, d_t)
2 $D \leftarrow$ SIMILARITYMETRIC(S)
3 $pred \leftarrow$ SVM(D)
4 **return** *pred*

without the need to exhaust resources by collecting and analyzing different variables. Data evolution patterns, or data pattern progress, are fundamental aspects of data behavior analysis as these patterns are the ones that provide the SVM with a strong basis to predict the data behavior that the sensed phenomena will exhibit next.

5.3.1 Augmented Candlestick Data Abstraction

In financial markets, the Japanese candlestick charting is a technique widely used for investment decision making [Nis01]. This technique facilitates the identification of patterns in price movements of currencies. In MWSN, we use candlesticks to extract features that describe the changes experienced by the sensed data in a user-defined time duration, d_t. In our BBTP approach, the candlestick abstraction technique is the foundation of the data behavior analysis. A candlestick extracts the first, last, minimum, and maximum sensed values registered during a time partition. Traditional augmented Japanese candlestick has a feature that also represents the number of transactions that took place during a time partition [CJKO15]. Differently from traditional augmented candlesticks, BBTP method substitutes the features related to the behavior of the financial markets with features that describe the behavior of the sensor node.

The features $Q(t)$ and $E(t)$ are the cardinality of the collected data and the Euclidean distance traveled during the time partition, respectively. $Q(t)$ serves as a means to compare the sampling interval stability between time partitions as a significant difference between the sampling intervals of two time series streams can influence the accuracy of the predictions made. Moreover, behavior extraction techniques usually make assumptions that are not met by real world scenarios (e.g., uniform sampling rate during each time partition). The feature $E(t)$ provides spatial information of the sensor node and is only added in applications where the captured phenomena depict spatio-temporal features, such as environmental monitoring applications [KXL+13]. A candlestick C at time partition t with duration d_t can be represented as follows:

$$C(t) = (F(t),\ X(t),\ N(t),\ L(t),\ Q(t),\ E(t)) \tag{5.1}$$

where:

$F(t)$: First sensed value.

$X(t)$: Maximum sensed value.

$N(t)$: Minimum sensed value.

$L(t)$: Last sensed value.

$Q(t)$: Cardinality of the collected data.

$E(t)$: Euclidean distance traveled during time partition.

To quantify the similarities between candlesticks, the extracted features must be comparable. A normalization process to allow this comparison must take place using the following:

$$A'(t) = \frac{A(t) - L(t)}{\sigma_L} \tag{5.2}$$

$$B'(t) = \frac{B(t)}{\sigma_B} \tag{5.3}$$

where the Equation (5.2) is used to normalize the first sensed value $F(t)$, the maximum sensed value $X(t)$, and the minimum sensed value $N(t)$ of the candlestick. $L(t)$ and σ_L are the last sensed values and its standard deviation, respectively. Equation (5.3) is used to normalize the cardinality of the total sensed values collected $Q(t)$, the Euclidean distances traveled by each sensor in one candlestick $E(t)$, and the last sensed values $L(t)$ where σ_B is their respective standard deviations. A sequence of multiple normalized candlesticks allow for a generalized visualization of behavior evolution over prolonged periods of time. The total number of candlesticks is denoted as n. A candlestick chart represented by a sequence of normalized candlesticks $S(t)$ is defined as:

$$S(t) = (C'(1), C'(2), ..., C'(n)) \tag{5.4}$$

Upon completion of the candlestick charts, the sensor discards the sensed data and stores only a data stream containing the sequence of normalized candlesticks. A description of the data abstraction process is showed in the Algorithm 4. Once data abstraction has been completed, the sensor node is ready to quantify the similarities between each candlestick by using the dynamic time warping technique.

Algorithm 4: Augmented Candlestick Data Abstraction

Input : Sensor data steams ($\{x_1, x_2, x_3, ..., x_n\}$), user-defined time duration (d_t)

Output: Stream of candlesticks (S')

1 $t_0 \leftarrow, t \leftarrow 1$
2 $t_f \leftarrow t_0 + d_t$
3 **while** $t_f \leq n$ **do**
4 $\quad S_F(t) \leftarrow x_{t_0}$
5 $\quad S_X(t) \leftarrow max(\{x_{t_o}, ..., x_{t_f}\})$
6 $\quad S_N(t) \leftarrow min(\{x_{t_o}, ..., x_{t_f}\})$
7 $\quad S_L(t) \leftarrow x_{t_f}$
8 $\quad S_Q(t) \leftarrow len(\{x_{t_o}, ..., x_{t_f}\})$
9 $\quad S_E(t) \leftarrow dist(\{x_{t_o}, ..., x_{t_f}\})$
10 $\quad t \leftarrow t + 1$
11 $\quad t_0 \leftarrow d_t * t$
12 $\quad t_f \leftarrow t_0 + d_t$
13 **end**
14 $S'_F(t) \leftarrow \frac{S_F(t) - S_L(t)}{\sigma_{S_L}}$ // From Equation (5.2)
15 $S'_X(t) \leftarrow \frac{S_X(t) - S_L(t)}{\sigma_{S_L}}$
16 $S'_N(t) \leftarrow \frac{S_N(t) - S_L(t)}{\sigma_{S_L}}$
17 $S'_L(t) \leftarrow \frac{S_L(t)}{\sigma_{S_L}}$ // From Equation (5.3)
18 $S'_Q(t) \leftarrow \frac{S_Q(t)}{\sigma_{S_Q}}$
19 $S'_E(t) \leftarrow \frac{S_E(t)}{\sigma_{S_E}}$
20 **return** S'

5.3.2 Similarity Quantification

We employ the Dynamic Time Warping (DTW) algorithm to measure the similarity between two consecutive candlesticks [SC07]. This step of the behavioral analysis compares the past and evolving patterns to aid in the prediction of future data trends. The DTW algorithm calculates the optimal alignment path between the individual elements, or attributes, in each candlestick to quantify their dissimilarity. The smallest path found between two candlesticks evaluates how different two candlesticks are. Two candlesticks that are identical will result in a DTW measure

of zero. The dissimilarity between two candlesticks is summarized in a single value, which favors a faster classification during the next step of BBTP and reduces the space complexity to the final form in our method. After the DTW is completed, the data is entirely abstracted.

The Euclidean distance is the distance measure used between two candlesticks and is the sum of the squared distances from the n-th attribute in one candlestick with the n-th attribute value in the other. A warp path $W = (w_1, w_2, ..., w_a)$ is constructed when comparing the attributes of the candlesticks. This warp path is of length a, where a is the number of total attributes used in the candlesticks. As in Figure 5.2 a warp path can be found from $D(F(t+1),\ F(t))$ to $D(E(t+1),\ E(t))$ between candlesticks $C(t)$ and $C(t+1)$, and is calculated upon completion of the filled cost matrix. The cost matrix is filled one column at a time starting from the bottom to the top, from the far left to the right column. The value of one cell in the cost matrix is calculated using the following:

$$D(i,j) \quad = \quad Dist(w_i, w_j) \ + \ min[D(i \ - \ 1, j), D(i, j \ - \ 1), D(i \ - \ 1, j \ - \ 1)]$$

where i and j represent each normalized data attribute F, X, N, L, Q, and E from the candlesticks $C(t)$ and $C(t+1)$. The warp path is calculated in reverse order starting from $D(E(t), E(t+1))$. A greedy search is performed that evaluates cells to the left, down, and diagonally to the bottom-left, similarly to the calculation of the cost matrix. Whichever of these three adjacent cells has the smallest value is added to the beginning of the warp path found so far, and the search continues from that cell. The search stops when $D(F(t), F(t+1))$ is reached. The minimum distance, between two consecutive candlesticks, $C(t)$ and $C(t+1)$, is defined as:

$$Dist(W) = \sum_{k=1}^{a} Dist(w_{ki},\ w_{kj}) \tag{5.5}$$

where $Dist(w_{ki}, w_{kj})$ is the dissimilarity between the two data point indexes, i and j, visually represented by one square in the cost matrix. The optimal path alignment or minimum-distance warp path is calculated by constructing a two-dimensional

$a \times a$ cost matrix D, where the value of D is the minimum distance warp path that can be constructed from two consecutive candlesticks. The value that truly describes the behavior in a sequence $S(t)$ is obtained when the DTW minimum distance measures have been calculated for all pairs of consecutive candlesticks. Lastly, for each candlestick, the DTW similarity measure between one candlestick and the next is stored. The candlesticks are discarded and only the DTW values are employed for training. The usage of DTW values facilitates the discovery in data evolution patterns to predict the behavior of future data trends.

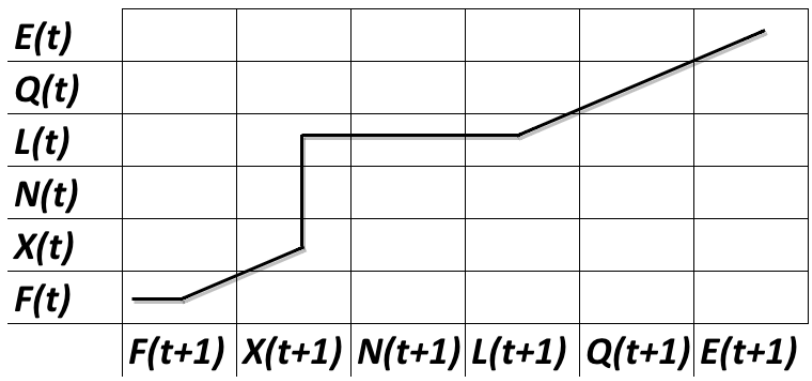


Figure 5.2: Cost matrix for Candlesticks C(t) and C(t+1) with a traced warp path

Algorithm 5: Dynamic Time Warping Similarity Measure

Input : Stream of n candlesticks (S)
Output: Stream of $n-1$ DTW values (D)

```
for i in {0, 1, 2, ..., n − 1} do
    for j in {0, 1, 2, ..., a} do
            // Note: a is the number of features in each candlestick
        for k in {0, 1, 2, ..., a} do
            cost ← Dist(C_i[j], C_{i+1}[k])          // From Equation (5.4)
            D ← cost + min[D(j − 1, k),
            D(j, k − 1), D(j − 1, k − 1)]            // From Equation (5.5)
        end
    end
end
return D
```

5.3.3 Data Behavior Learning and Prediction

To learn from the sensor's data behavior and subsequently perform predictions, BBTP uses a multi-class Support Vector Machine (SVM) [WX14]. As studied by various authors [WZWM17, MHF07, TNK17], SVM has good performance for classification problems when compared with other techniques, including Random Forest, HMM and K-NN. Support Vector Machines tend to be less prone to overfitting problems, a desirable quality in MWSN given that the presence of noisy data is common in these types of networks. Overfitting happens when a model learns the details and noise in the training data to the extent that it negatively impacts the performance of the model on new data. In other words, when a model overfits, the model recognizes the noise in the training data and learns it as principle.

The SVM step of the BBTP technique utilizes the DTW output and its corresponding labels to learn from the behavior extracted from the already abstracted data. The labels are assigned depending on the trend between two DTW values, or three consecutive candlesticks. There are three possible classes or trends: a negative or downward trend, zero or flat trend, and a positive or upward trend. For each dissimilarity measure between DTW values y_{t-1} and y_t, the label l_t is determined as follows:

$$l_t = \begin{cases} -1, & y_{t-1} > y_t \\ 0, & y_{t-1} = y_t \\ 1, & y_{t-1} < y_t \end{cases} \tag{5.6}$$

Once the dissimilarity measures have been classified, the SVM is trained. Given training data $\{(y_1, l_1), ..., (y_n, l_n)\} \in \mathbb{R} \times \{-1, 0, 1\}$, where $t \in \{1, 2, ..., n\}$. Our goal is to solve the following primal problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{t=1}^{n} (\zeta_t) \tag{5.7}$$

subject to $l_t(w^T \phi(y_t) + b) \geq 1 - \zeta_t$, $\zeta_{\geq} 0$, and $t = 1, 2, ..., n$. Its dual is

$$\min_{\alpha} \frac{1}{2} (\alpha)^T Q(\alpha) - e^T(\alpha) \tag{5.8}$$

subject to $l^T(\alpha) = 0$, $0 \leq \alpha_t \leq C$, and $t = 1, 2, ..., n$ where y and l are the dissimilarity measures and the corresponding trend classes, respectively. w is a linear combination of the training patterns, and $C > 0$ is the upper bound. ζ_t is a slack variable that allows for errors and approximation in case the above problem is unfeasible. Q is an $n \times n$ positive semidefinite matrix, $Q_{ij} \equiv k(y_i, y_j) = \phi(y_i)^T \phi(y_j)$ is the kernel of the function ϕ, which is employed to map the training vectors into feature space, and here i and $j \in t$, but $i \neq j$. Furthermore, e is the vector of all ones, the threshold b is computed to satisfying the Karush-Kuhn-Tucker (KKT) conditions [**?**], and α_t, α_t^* are weight Lagranian multipliers employed to perform the dualization of the primal problem. Finally, the decision function is returned by:

$$\text{sgn}(\sum_{t=1}^{n} l_t \alpha_t K(y_t, y) + \rho) \tag{5.9}$$

where $K(y_t, y)$ represents the evaluating kernel functions. This function returns a predicted data trends. As the SVM technique is trained given dissimilarity measures, the predicted trend is dependent on the abstracted behavior rather than just the behavior of raw sensed values.

5.4 Experimental Results

To assess the performance of our proposed method, we conducted simulations using the Python programming language and two real-world datasets, OpenSense [LFS+12] and Physionet MIMIC II [SVR+11]. To contrast the results of BBTP, we have selected four well-known classification methods widely employed for predictions [KAA18]: Random Forest, Decision Tree, K-Nearest Neighbor (K-NN), and SVM without our behavior abstraction step.

As a prediction accuracy metric, we have employed the statistical metrics recall, precision and accuracy score [SL09]. Recall measures the ability of a classification model to identify all relevant instances, while precision measures the ability to return only relevant instances. While recall measures the ability to find all pertinent instances of a dataset, precision measures the proportion of the data points our

model says was relevant truly were relevant. The Table 5.1 shows the confusion matrix for the four possible outcomes during prediction where precision and recall are mathematically formulated as:

$$precision = \frac{true\,positives}{true\,positives + false\,positives} \tag{5.10}$$

$$recall = \frac{true\,positives}{true\,positives + false\,negatives} \tag{5.11}$$

		Actual	
		Positive	**Negative**
Predictions	**Positive**	True Positive	False Positive
	Negative	False Negative	True Negative

Table 5.1: Confusion Matrix for Binary Classification

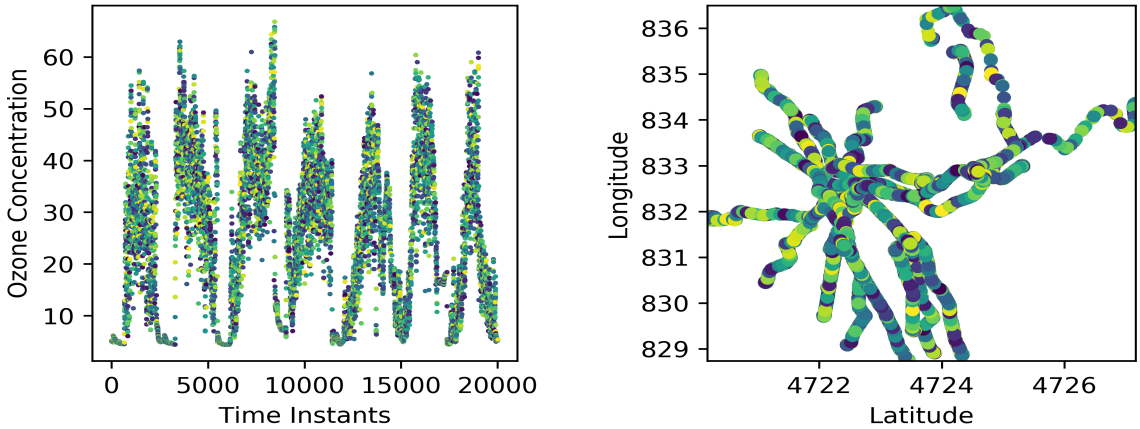
In addition, we computed the accuracy score, which evaluates the subset of predictions which were labeled exactly the corresponding set of true labels. In other words, if p'_i is the predicted value of the i-th sample and p_i is the corresponding actual value, then the fraction of correct predictions over n_{trends} the accuracy score. This is defined as:

$$Accuracy(p, p') = \frac{1}{n_{trends}} \sum_{i=0}^{n_{trends}-1} 1(p'_i = p_i) \tag{5.12}$$

5.4.1 OpenSense Dataset

In the OpenSense project, sensors were placed on ten trams traveling through the city of Zurich. The installed sensors collected temperature, humidity, and the ozone concentration levels along with their longitude and latitude from May 31st, 2013 18:40:00 to June 7th, 2013 20:06:40. These attributes have spatio-temporal correlation [KXL+13]. As a result, it contains finite-time stability and will not change abruptly. Making the use of this dataset is suitable for our BBTP technique as it is an evolving data stream. We specifically employed our BBTP method utilizing the

ozone concentration data values, along with the latitude and longitude points. The total amount of collected data in this simulation is 124,815 samples of ozone concentration levels, latitude, and longitude points. The average amount of data collected from each tram is 11,347 samples. Figure 5.3 contains a visual representation of all the collected ozone concentration levels and latitude/longitude points over time.



(a) Collected Ozone Concentration Levels

(b) Trajectories Coverage

Figure 5.3: OpenSense Dataset

For this simulation, it is assumed that outliers have been eliminated. All data contained in the sensor internal memory is considered valid and to be used for the future data prediction. Sensors have *a priori* knowledge of the area in which the are deployed. With every sensed value collected, the sensor will register timestamp and its location (latitude and longitude) information. Once the data has been collected, the timestamps, ozone levels concentrations, and location points are used to model the behavior of the data through our data abstraction step. Once data abstraction has been performed, the reduced data is forwarded to the IoT device. The IoT device learns the behavior and performs the future trend prediction of ozone concentration levels for the specific route. The timestamp represented in each candlestick was 10 minutes in duration (d_t = 10 minutes). In this study, since environmental data depicts a spatio-temporal relation [KXL+13], the attributes extracted to construct candlesticks are the first (F), last (L), minimum (N), and maximum (X) sensed values. In addition, the cardinality of the collected data per candlestick (Q), and

the Euclidean distance traveled during the time partition (E) were included as augmented features in our candlestick data structure. After the candlestick charts were constructed, behavior dissimilarity measures were computed resulting in 3,069 values employed for training the model. For the contrasting methods, the training data size employed was the original 62,409 ozone concentration values. Half of the data was used for training, and the other half was used for testing. In this application, we simulated this by training for 24 hours and testing for 24 hours at a time.

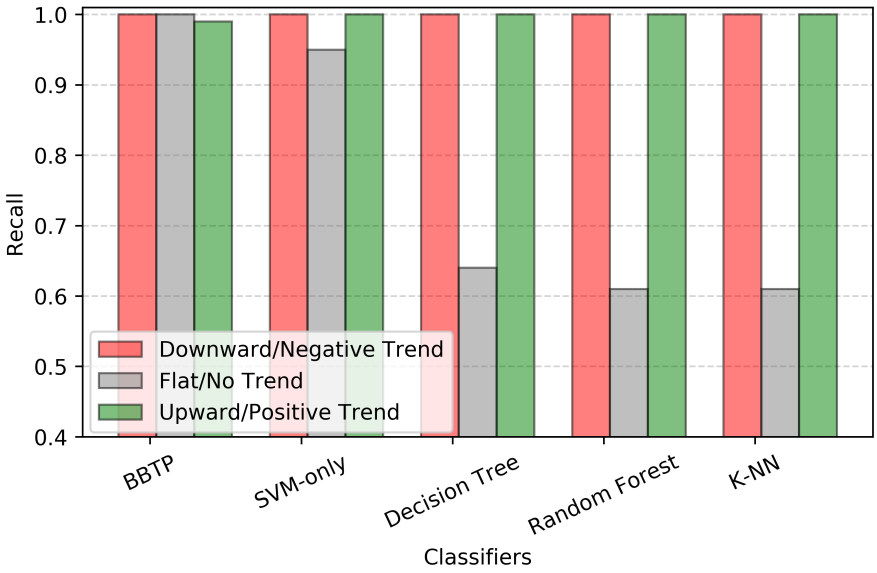


Figure 5.4: Recall in OpenSense Simulation

The ability to predict upward trends, downward trends and flat trends, and the over all accuracy of the predictions made for all studied methods, can be examined in Figure 5.4, 5.5 and 5.6. It can be observed that BBTP outperformed SVM, Random Forest, Decision Tree and K-NN. Moreover, it is notable that SVM showed superior performance against Random Forest, Decision Tree and K-NN, which confirms the selection of SVM as the appropriate classification method employed in the final stage of BBTP. Although all classifiers had a good performance and that BBTP performance was slightly better than SVM, it is important to note that the amount of data employed to train the model in BBTP represents 4.92% of the training data

utilized by the other techniques as observed in Table 5.2. Therefore, we can draw the conviction that BBTP's data abstraction and behavior characterization strategies

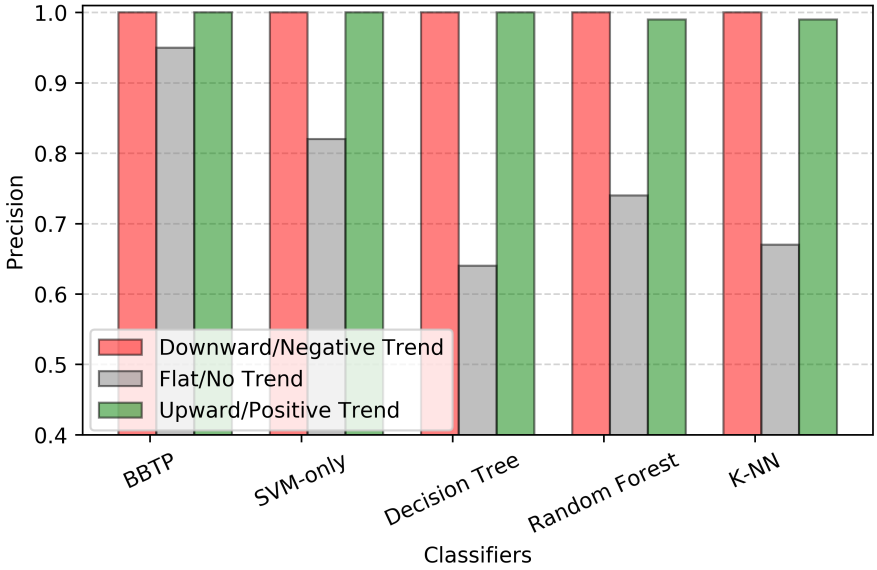


Figure 5.5: Precision in OpenSense Simulation

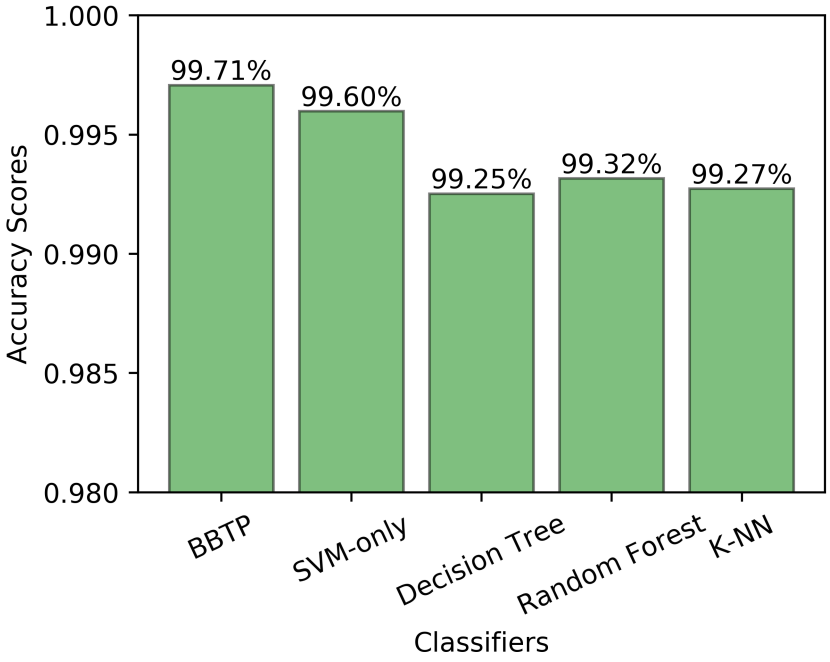


Figure 5.6: Prediction Performance in OpenSense Dataset

effectively model the behavior of evolving data streams.

Prediction Technique	Accuracy Score	Training Size
BBTP	0.9970	3,069
SVM-only	0.9964	62,409
Decision Tree	0.9925	62,409
Random Forest	0.9935	62,409
K-NN	0.9972	62,409

Table 5.2: Accuracy Scores for OpenSense Dataset

5.4.2 PhysioNet MIMIC II Dataset

PhysioNet MIMIC II database used in our simulation contains 4,458 vitals records from 3,704 adult Intensive Care Unit (ICU) patients and 249 neonates. For this study, we have selected the time series of vital signs sampled. The hemoglobin saturation levels (SpO_2) were sampled per minute for thirty different patients during different dates from 2009 to 2017. The data recorded for each patient ranged from one to fifteen days, resulting in a with a total 122 days of total data. The average amount of data collected from each patient was 5,778 samples. In total 216,767 samples were collected, and 173,312 samples remained after the outliers were removed. Figure 5.7 shows the SpO_2 values for a sample patient. Most of the patients contained values that behave similarly. Although patients' vitals data had an evolving data behavior, it is not as evident in all patients, as in the air quality data. It is common for physiologic time series to be interrupted or changed occasionally during recordings of such long duration [GAG+00].

The time represented in each candlestick was 15 minutes in duration ($d_t = 15$ minutes). In this study, the sensed data does not depict spatial dependence and for that reason the Euclidean distance augmented feature was not included in the data behavior analysis. The attributes extracted to construct candlesticks are the first (F), last (L), minimum (N), and maximum (X) sensed values. In addition, the cardinality of the collected data per candlestick (Q). After the candlestick charts were constructed, behavior dissimilarity measures were computed resulting in 9,241

values employed for training the model. For the contrasting methods, the training data size employed was 138,649 values. A total of 2,310 trends were predicted employing each of the tested methods. Due to the increased amount of data available and the lower overall sensed data variance, all techniques tested performed with a

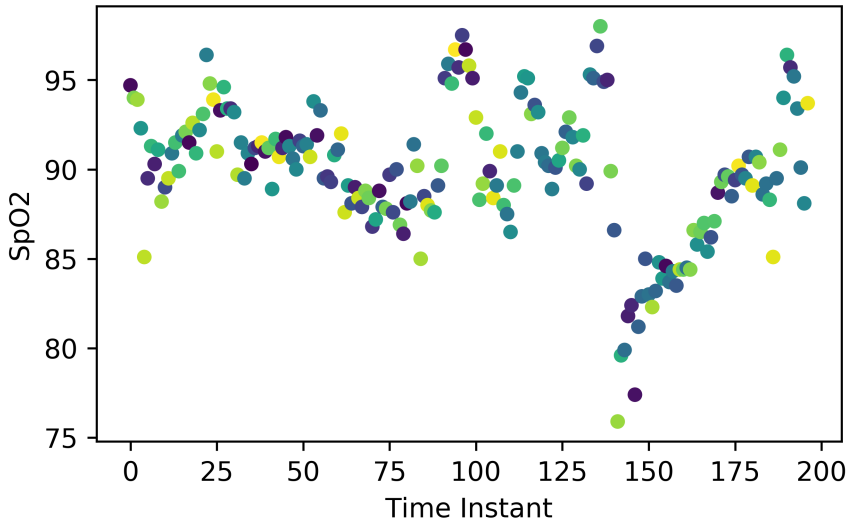


Figure 5.7: Sample Patient from PhysioNet MIMIC II Dataset

Figure 5.8: Recall in PhysioNet MIMIC II Simulation

Figure 5.9: Precision in PhysioNet MIMIC II Simulation

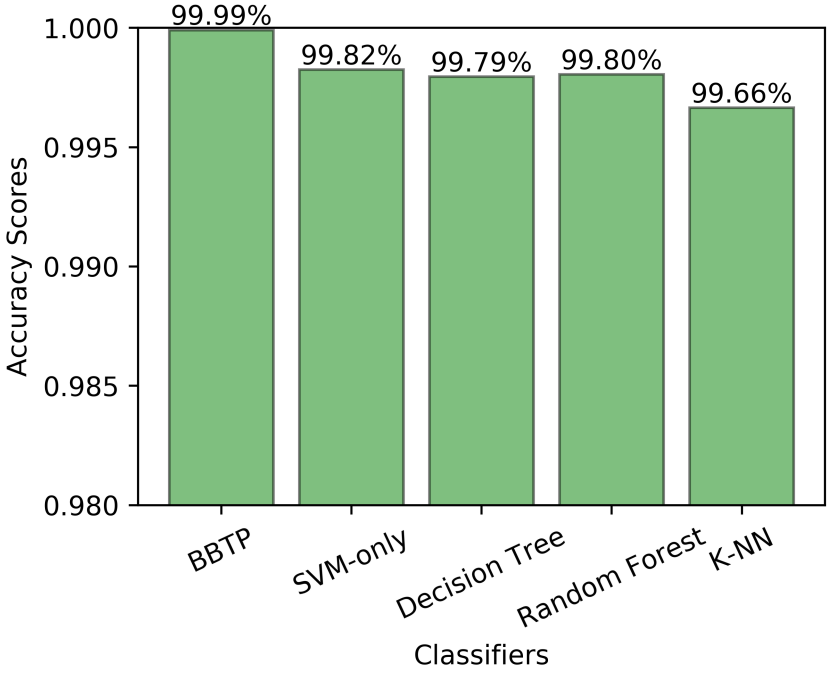


Figure 5.10: Prediction Performance in PhysioNet MIMIC II Dataset

higher accuracy compared to the environmental simulation. However, BBTP still employed less than 7% of the data used by other methods for model training and it showed comparable recall and precision scores and a higher accuracy score as showed in Table 5.3 and Figures 5.8, 5.9 and 5.10.

Prediction Technique	**Accuracy Score**	**Training Size**
BBTP	0.9999	9,241
SVM-only	0.9982	138,649
Decision Tree	0.9979	138,649
Random Forest	0.9980	138,649
K-NN	0.9966	138,649

Table 5.3: Accuracy Scores for PhysioNet MIMIC II Dataset

5.5 Summary

Our Behavior-Based Trend Prediction approach is a novel data trend prediction technique designed to model the behavior of evolving data streams in MWSN. Our method reduces the space complexity in sensor nodes and exploit the increasing IoT technology benefits. The aftereffect of reducing the size of the data at the node level also corresponds to a reduction of the network traffic, which as well, may lead to fewer message collisions and re-transmissions.

BBTP's ability to effectively model evolving data streams' behavior has been demonstrated through its simulations in two real-world datasets. During the candlestick data abstraction, together with the sensing data, BBTP captures meaningful information that describes the true conditions of the real-world to model the sensing data behavior. This information includes the sampling interval stability and the distance traveled by the sensor node between consecutive time partitions. The main virtue of the dynamic time warping lies in its capability to encapsulate the behavior of the data together with the real world conditions in a single value. This behavior characterization makes it possible to discard large amounts of data without losing the information about behavior over time. In addition, the use of the behavior dissimilarity measure as input to train the SVM enables the use of a tiny fraction

of data compared to the original sensed data, while still maintaining competitive results.

BBTP had better efficacy than SVM method without data abstraction, even when SVM employed up to 95% more data to train the model. Moreover, BBTP's superior performance is justified by its behavior-based learning principals, rather than learning from raw data. This quality makes BBTP suitable for MWSN, where there is limited memory, low computational capability, and small or irreplaceable power sources. Overall, BBTP shows diverse applicability in mobile wireless sensors in the field of Internet of Things.

CHAPTER 6

DATA CLEANING DISTRIBUTION IN EH-MWSN

The use of energy-harvesting technologies in mobile wireless sensor networks (MWSN) delivers a promising opportunity to mitigate the limitations that irreplaceable energy sources impose over conventional MWSN. Existing energy-efficient methods that exploit the benefits of energy-harvesting technologies focus on increasing the uptime of individual sensor nodes; however, they lack planning in the survivability of the network as a whole. Moreover, the existing methods do not consider the consequences that dirty data can have on real-time decision-making applications.

In this chapter we propose an Leontief Data Cleaning Distribution Strategy (Leontief-DCD), an economic-based method designed to distribute the data cleaning workload in energy-harvesting MWSN powered by predictable energy sources, such as solar energy. The resulting data cleaning distribution strategies computed aim to increase network uptime and the quantity of data that is run through data cleaning processes. Our method creates interdependencies among sensor nodes to predict the required cooperation from each node in the data cleaning process benefiting the network as a whole, rather than only individual sensors. Furthermore, our results show that when employing our method to distribute data cleaning workload in highly dirty, real-world datasets in scenarios with high and low energy, our method increased the number of data samples engaged in data cleaning processes by up to 25.57%, the count of active sensor nodes by up to 44.01%, and the network overall well-being by up to 55.42%. This chapter is organized as follows: The Section 6.1 offers an introduction to energy harvesting MWSN. The problem statement can be reviewed in Section 6.2. The Section 6.3 present our proposed method. The evaluation of results are described in Section 6.4. Lastly, Section 6.5 presents a summary.

6.1 Introduction

The limited energy resources that characterize MWSN is a major bottleneck for applications in this type of networks. However, through energy-harvesting technologies, nodes are in theory enabled to have infinite lifetime when in Energy Neutral Operation (ENO) [KHZS07]. Formally defined, a node in ENO state harvests more energy than it consumes [AMAT+18]. In other words, its energy consumption rate is always less than its harvesting energy rate. Although sensors' lifetime is not a critical problem in energy-harvesting mobile wireless sensor networks (EH-MWSN), network uptime can be affected. Generally, in EH-MWSN, when a node does not possess enough energy to perform its normal functions, it adopts energy-saving strategies and becomes operational only when its onboard residual energy reaches a predefined threshold. These energy-saving strategies, together with the reduction of the network's uptime, may limit a sensor network's functionality, including the ability to sustain data accuracy to be employed in decision-making applications.

As discussed in chapter 2, substantial attention has been placed in methods involving the management of energy with the purpose to reduce network downtime [FD11, GHZH14, ZCZ+16, KHZS07, VGB07, ZSA11, SSCS17, TAH+15, Cui18]. Nevertheless, limited attention has been placed on increasing data quality. Moreover, data collected in EH-MWSN has an exponential growth, and its mobility challenges the accuracy of data due to sensor isolation, short-term sensor connectivity and data collision[PGWC16, SBB13b]. All these difficulties result in high volumes of missing, noisy or duplicated data.

With the large quantities of dirty data provided by mobile nodes, data cleaning becomes critical due to its negative effects on data mining, machine learning models, and decision-making applications [QWLG18, PGWC16]. For this reason, we focus on helping sensor nodes increase data quality by distributing the data cleaning among sensor nodes based on their onboard residual energy and quantity of dirty samples while seeking to drive the network to a neutral operation state. It should be noted that ideally, a network is described to be in neutral operation, Network Neutral Operation (NNO), when all the sensor nodes that compose the network are in ENO state. We establish the future NNO state by employing a simple regression

and the sensor's onboard residual energy historical data to predict the energy at the end of the following time slot. Finally, a workload distribution strategy is computed based on the NNO state of each cluster.

In this method we apply the Leontief Input-Output model to achieve NNO by distributing the data cleaning workload, without affecting network uptime in EH-MWSN. Economic theories have been successfully solving problems in the world's economy, and in economy it is assumed that individuals are rational and have limited resources. For this reason, resources must be carefully allocated for maximum benefits. Likewise in EH-MWSN, sensor nodes are considered rational and have limited energy; therefore must use it carefully to avoid depletion. Additionally, in economics individuals make decisions to obtain the most happiness at the least cost. Similarly, sensors make decisions driven by self-interest to sustain high-quality of service (QoS) and increase lifetime/uptime.

The Leontief Input-Output model analyzes the production and consumption in an economy by quantifying the interdependencies between different sectors of an economy [Leo86]. Correspondingly, in EH-MWSN we use this model to analyze the relationship between production and consumption in a designed data cleaning workload distribution among the nodes of a network. In the Leontief model, economies are divided into sectors, compared to EH-MWSN, where networks are composed of nodes. Sectors produce and consume products/services that are used by other sectors; in a like manner, sensor nodes produce and utilize data or other tasks that are employed or delivered by other sensors. Examples of these tasks include data forwarding, data cleaning and information sharing. In economics, this model assumes that the demand is met without surplus or shortage, while in EH-MWSN the energy the network uses for data cleaning is provided by the sensors of the network. Overall, in both domains while the strategy is centrally planned, the execution of the tasks is performed in a distributed manner.

6.2 Problem Statement and Assumptions

The goal of our work is to develop a data cleaning workload distribution system for EH-MWSN. The aim of this method is to ensure the availability of accurate data while increasing network uptime in networks composed of nodes possessing heterogeneous functions and capabilities. We consider an EH-MWSN composed of n nodes deployed in a wide area where sensors communicate among themselves. Nodes are embedded into vehicles and/or devices carried by people and move in a determined pattern. Figure 6.1 shows an example of an EH-MWSN deployed in a military environment. Deploying soldiers can be risky and dangerous. For this reason, in this scenario, the middle area is patrolled by autonomous mobile nodes to gather and distribute information to be employed in different applications including perimeter surveillance and protection, nuclear, chemical, and biological attacks detection, and missile monitoring [AFS17]. The availability of high-quality data in EH-MWSN for military applications is imperative as it can decrease fatality rate. Due to their mobility, some nodes received more solar energy than others throughout their trajectories. The prediction of the energy to be harvested at the node can help nodes adjust their functions to achieve individual nodes ENO.

Additionally, the energetic cost of data cleaning can be elevated for sensor nodes with large amounts of data streams [SZ16]. Although, energy harvesting-enabled nodes promise to deliver infinite lifetime when deployed in environments with a constant energy supply, the heterogeneity and mobility of the sensors add challenges that severely affect the collection of high-quality data [SBB13b] and uptime extension. To increase network uptime while ensuring high-quality data, it is necessary to reach NNO instead of seeking to achieve individual node ENO. As a result, energy-harvesting prediction can support the process of data cleaning workload distribution, which in return can help to accomplish NNO while propelling the availability of high-quality data. It is assumed that sensors are mobile, cooperative and collaborative. In addition, sensors measure and store data in their internal memory, and each one has an internal pre-process for detection and elimination of dirty data, resulting in missing data. Since the second greatest energy consumer is computational operations in in-network processing and, in EH-MWSN, the availability of a base

station or a front-end for processing of large volumes of data is unlikely, the cost of in-network data is elevated [AQAKS17]. Therefore, the in-network data cleaning workload will be distributed.

The energy expenditure and energy-harvesting rates differ for each sensor node and are dependent on each sensor operation and energy supply availability. We borrow concepts from the efficient market hypothesis to create an energy profile that can allow sensors to predict their future onboard residual energy. This financial theory states that the price of a financial security fully reflects all relevant available information [Fam66]. Likewise, we assume that in EH-MWSN the onboard residual energy on sensor nodes reflects all available relevant factors (i.e. environmental conditions, energy harvesting system architecture and sensor node functions) that affect it. In addition, we consider that all onboard residual energy data is valid and to be employed for future energy values predictions.

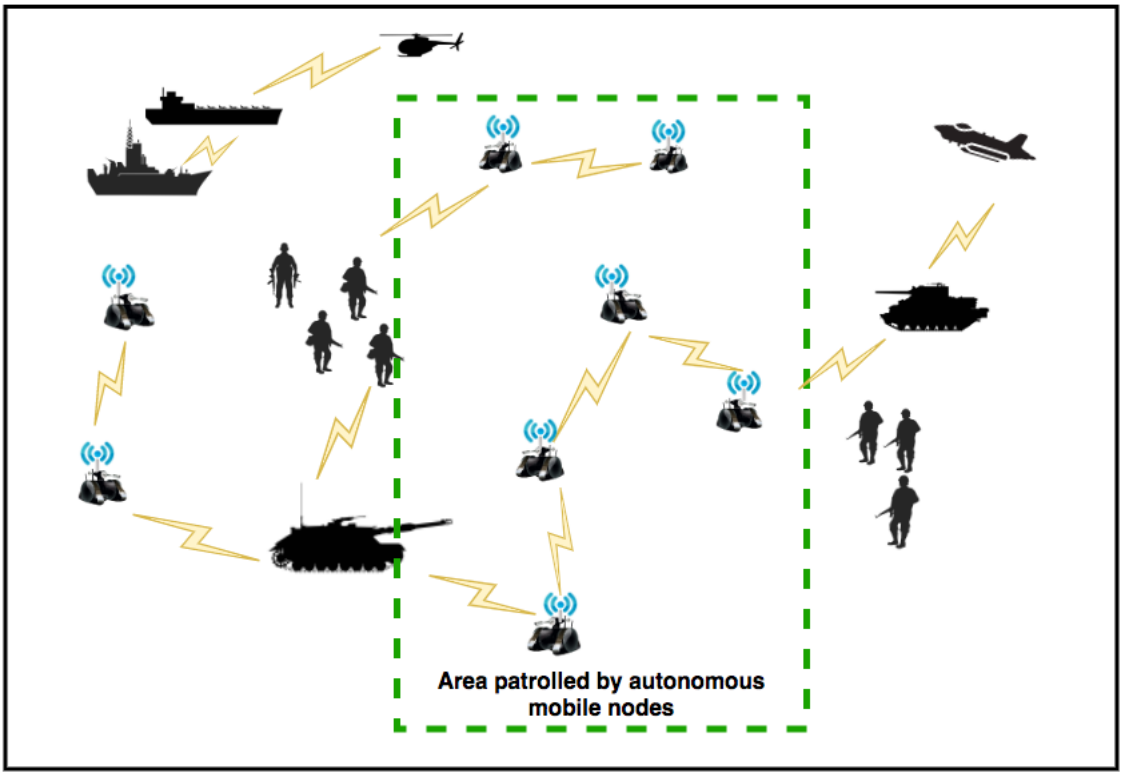


Figure 6.1: Example of EH-MWSN in Military Application

We assume that at the end of each iteration, sensors form clusters and select a cluster-head. Then, sensors send information related to their location, actual and predicted residual energy, as well as the amount of dirty data to be cleaned to the cluster-head. The cluster-head performs the data cleaning workload distribution, and the data cleaning is completed by each node as instructed. Moreover, sensors are assumed to have *a priori* knowledge of the area in which they are deployed.

6.3 Distributed Data Cleaning Strategy

In this section, we present our methodology for analyzing the energy and dirty data input-output relationship employed to determine how to distribute the data cleaning workload and achieve NNO. Sensor nodes within a EH-MWSN are heterogeneous in their construction and functions, and often seek to reach ENO individually, risking the network survivability [AMAT+18]. Our technique keeps sensor nodes from having to choose between data accuracy and NNO.

To understand how the Leontief Input-Output model is applied to EH-MWSN, it is important to notice that the input-output relationship of energy and dirty data in EH-MWSN are similar to supply and demand in economics. In economics, the economy is composed of sectors which produce and consume products/services that are used by other sectors. The output is the quantity of goods/services produced in a given period of time by an industrial sector, to be consumed for further production, and the input is what is used to generate output [Leo86]. The sensor nodes that belong to an EH-MWSN behave similarly to industrial sectors. Sensor nodes within a network can cooperate by generating data/providing services that can be used by other sensor nodes. In the context of our research, sensor nodes can cooperate by cleaning data of other sensors to be employed in decision making.

6.3.1 Energy Profile

To predict sensor nodes' future onboard residual levels, we construct an energy profile for each sensor node employing sensors' onboard residual energy historical data. To approximate the onboard residual energy at the beginning of the next

iteration we employ linear regression [MPV12]. We use this data to predict the future onboard residual energy, Z, by finding the linear regression line in the form of $Z = mx + b$ where x is the independent variable mapping to the times at which the onboard residual energies were collected, m is the slope, and b is the y-intercept. m is calculated as follows:

$$m = \frac{u \sum xy - (\sum x)(\sum y)}{u \sum x^2 - (\sum x)^2} \tag{6.1}$$

and

$$b = \frac{\sum y}{u} - m\frac{\sum x}{u} \tag{6.2}$$

where y is the collected onboard residual energies and u is the number of samples collected per iteration.

6.3.2 Data Cleaning Distribution Strategy

Our Leontief-DCD method is divided in two phases. The first phase determines whether or not the sensors are in ENO and establishes the desirable state of the cluster, NNO state. The second phase constructs a fair data cleaning workload distribution strategy. In this step, Leontief-DCD takes in consideration sensors' onboard residual energy and the amount of dirty data.

Desirable Future State

In EH-MWSN, sensors need to be in ENO to theoretically achieve an infinite lifetime. We define ENO as the most desirable state, where the harvesting energy rate is greater than or equal to the energy expenditure rate. In other words, sensors in ENO state consume less energy than what they harvest. The identification of sensor nodes that depict this positive increment in energy allows us to assign data cleaning workload to this sensor nodes without compromising their uptime. To determine ENO state, for each sensor node, we compare the current onboard residual energy and the final onboard residual energy of the subsequent time slot, $E_{(n,t)}$ and $Z_{(n,t)}$, respectively. If the energy-harvesting rate is greater than its consumption rate, such

Algorithm 6: Determine sensors that are ENO for a singular cluster. Completed by each cluster head node.

Input : All sensors in the network cluster (Y), each sensor's current (E) and predicted (Z) onboard residual energies at time t

Output: Set of sensors who are in the ENO state (ENO_v), set of sensors who are not in the ENO state (ENO_i)

```
for n in Y do
    if E_(n,t) < Z_(n,t) then
        ENO_v ← n                          // Sensor is in ENO state
    else
        ENO_i ← n                          // Sensor is not in ENO state
        E_(n,t) ← Z_(n,t)                  // Update onboard residual energy to the
            predicted residual energy
    end
end
return (ENO_v, ENO_i)
```

that $E_{(n,t)} < Z_{(n,t)}$, the sensor node is in ENO state. Similarly, if $E_{(n,t)} \geq Z_{(n,t)}$, the energy consumption rate is higher than the harvesting rate, so the sensor is not in ENO state. This procedure can be visualized in the Algorithm 6.

Fair Cleaning Strategy

The fair cleaning strategy determines the data cleaning workload that each sensor node in ENO will need to complete in order to fulfill NNO. This strategy creates interdependencies among sensor nodes part of the cluster, that constitutes an alliance to reduce the amount of dirty data while increasing the network uptime. To compute the fair cleaning strategy, we build a fair cleaning workload assignment table in which each sensor node will use the predicted future onboard residual energy, $Z_{(n,t)}$ if its energy operational state is not in ENO. Correspondingly, if the sensor operational state is in ENO, sensors will use the current onboard residual energy, $E_{(n,t)}$. This procedure will aid sensors out of ENO state to recover and avoid energy depletion by reducing the workload assigned to sensors with low residual energy. On

the other hand, if the predicted future onboard residual energy $Z_{(n,t)}$ is less than the computed threshold, the current cleaning assignment for the sensor is removed and its workload is evenly distributed among the remaining sensor nodes. This method is highly beneficial for sensors with alarming low onboard residual energy and high volumes of dirty data. The energy level threshold is computed as:

$$\psi = min(\omega, \bar{E}_{low}) \tag{6.3}$$

where ω represents the energy level where sensors are normally sent to sleep mode and $\bar{E}_{low}$ is the average energy levels for half of the sensors with the lowest onboard residual energy.

In this stage, we establish a fair cleaning strategy by distributing the cleaning workload based on their input-output relation, that is the onboard residual energy available for the cluster and the number of dirty data required to be cleaned within the cluster. We state that a sensor will clean a percentage of its dirty samples proportional to the percentage of the sensor's onboard residual energy, plus a potential additional assignment from other nodes. In addition, sensors with alarming low energy levels and high volumes of dirty data are not assigned a cleaning load if their energy level is below the threshold, ψ.

Sensor	1	2	...	n	ENO_i
1	D_1	S_{12}	..	S_{1n}	O_1
2	S_{21}	D_2	...	S_{2n}	O_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	S_{n1}	...	...	D_n	O_n

Table 6.1: Fair Cleaning Workload Assignment Table

Table 6.1 shows the fair cleaning strategy table computed by the cluster-head where the calculated values of each element represent the data cleaning workload assigned to sensor i from sensor j. Only sensor nodes in ENO are assigned data cleaning workload, and each row corresponds to a sensor that is capable of cleaning

dirty values, $n \in ENO_v$. The last column corresponds to an additional workload assigned to sensor n from those sensors that are not capable of cleaning, $m \in ENO_i$.

A sensor is assigned to clean a portion of its own dirty data in addition to a portion of dirty data belonging to other sensor nodes depending on the amount of onboard energies per sensor in each cluster. As a result, there are cases 3 during the data cleaning workload distribution: (1) a sensor i is cleaning its own data, (2) a sensor j's dirty data is be cleaned by sensor i, or (3) a sensor i does not clean any data because it is not in ENO. For the first case, we use Equation 6.4 where for each node i, we multiply the number of its dirty samples (d_i) by its onboard residual energy level ($E_{(i,t)}$).

$$D_i = d_i * E_{(i,t)} \tag{6.4}$$

For the second case, we use the Equation 6.5 where the energy level of the sensor i is divided by the total energy levels of the sensors with cleaning assignment within the cluster, excluding sensor j. It is important to note that the onboard energy level of a sensor is a percentage based on the sensor with the highest battery capacity, resulting in values between $[0, 1]$ and, therefore, S_{ij} will never be negative.

$$S_{ij} = \frac{E_{(i,t)}}{\sum_{n \in ENO_v \setminus \{j\}} E_{(n,t)}} (d_j - E_{(j,t)} * d_j) \tag{6.5}$$

For the final case, we use equation 6.6 to compute the cleaning workload assigned to sensor i belonging to the group of sensors with energy levels below the threshold, ENO_i. It is the energy level of the sensor i divided by the total energy levels of the sensors with cleaning assignment within the cluster, multiplied by the summation of dirty samples belonging to the sensors with energy levels below the threshold, ψ.

$$O_i = \frac{E_{(i,t)}}{\sum_{n \in ENO_v} E_{(n,t)}} \sum_{m \in ENO_i} d_m \tag{6.6}$$

Note that ENO_v and ENO_i are the sets of sensors with and without their energy levels above the threshold, ψ. In other words, the set of sensors that are capable of cleaning dirty data values and the set of sensors who are not capable. Therefore,

Algorithm 7: Strategic Data Cleaning Assignment T

Input : Sets containing energy operational state for all sensors in the cluster Y (ENO_v and ENO_i), each sensor's current energy (E), and their number of dirty samples (d) at time t

Output: Strategic data cleaning workload assignments (T)

```
// Complete fair cleaning workload assignment table
for i in ENO_v do
    for j in ENO_v ∪ {|ENO_v| + 1} do
        // Extra column for those sensors that are not capable of cleaning
        if i = j then
            F[i, j] ← Equation (6.4)                    // Use D_i
        else if i ≠ j: E_(i,t), E_(j,t) > ψ then
            F[i, j] ← Equation (6.5)                    // Use S_ij
        else
            F[i, j] ← Equation (6.6)                    // Use O_i
        end
    end
end
// Compute the total cleaning workload assignment for each sensor
for i in ENO_v do
    T[i] ← ∑ F[i]                    // Sum of all elements in a row i
end
return T
```

we know that $ENO_v \cap ENO_i = \emptyset$ and $ENO_v \cup ENO_i = Y$ where Y is the set of all the sensors in a cluster at the end of a given time interval.

The total cleaning workload assignment for each sensor is the sum of all the elements of its corresponding row in the Fair Cleaning Workload Assignment table. This procedure is depicted in Algorithm 7. It represents the required number of samples that each node needs to clean in order to achieve the NNO state of the

cluster Y, NNO_Y, and is computed as:

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_i \\ \vdots \\ T_n \end{bmatrix} = \begin{bmatrix} D_1 + \sum_{j \in ENO_v \setminus \{1\}} S_{1j} + O_1 \\ D_2 + \sum_{j \in ENO_v \setminus \{2\}} S_{2j} + O_2 \\ \vdots \\ D_i + \sum_{j \in ENO_v \setminus \{i\}} S_{ij} + O_i \\ \vdots \\ D_n + \sum_{j \in ENO_v \setminus \{n\}} S_{nj} + O_n \end{bmatrix} = NNO_Y \tag{6.7}$$

Once the cleaning workload assignments to accomplish NNO_Y have been computed, the cluster-head informs the sensor nodes of the data cleaning distribution strategy. It is then that sensors perform the cleaning and forward the cleaned data to the corresponding recipients. Sensors continue their normal operations until the next data cleaning period.

6.4 Performance Evaluation

To evaluate the performance of Leontief-DCD, we conducted simulations using Python and the Bonnnmotion Mobility Scenario Generation and Analysis Tool to generate sensor mobility. Also, we employed real-world datasets provided by The National Renewable Energy Laboratory, NWTC M2, Solar TAC, SRRL BMS, VTIF RSR and LRSS [NRE], and referred to as Region A to E, respectively. From these datasets, sensors collected irradiation at five different nearby locations from April 1st to May 30th, 2013 and from December 15th, 2012 to January 15th, 2013. This collected irradiance data was used in two scenarios, along with the location information, to be employed for energy harvesting/consumption profile creation when evaluating the performance of our proposed method. We used these two days specifically as they were different in terms of overall irradiation values collected. The two provided scenarios allow us to evaluate our technique on a sunny day and on a cloudy day.

Figures 6.2 and 6.3 show the collected irradiance values per region for a cloudy and a sunny day, respectively. We can see, though they exhibit similar patterns,

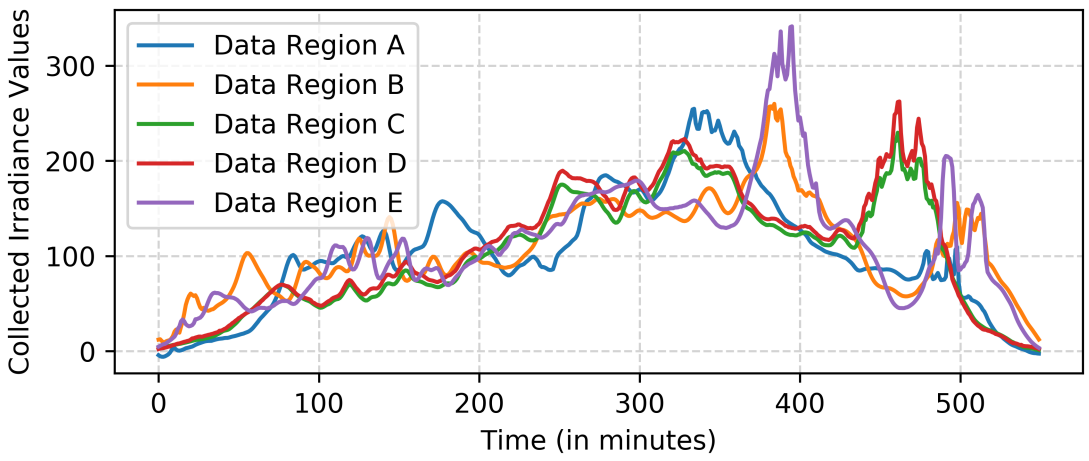


Figure 6.2: Collected irradiance values per region on a cloudy 9-hour day

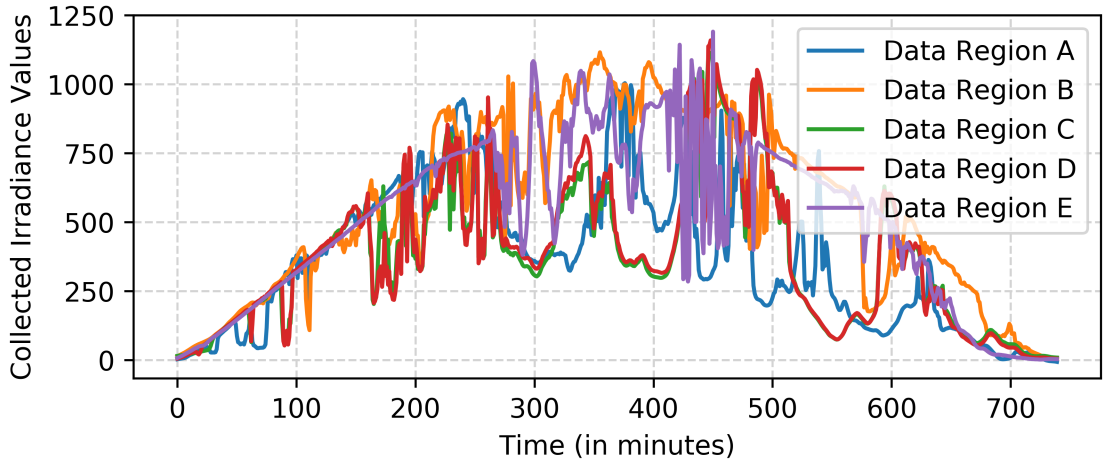


Figure 6.3: Collected irradiance values per region on a sunny 12-hour day

the cloudy day results in irradiance values that are significantly lower than the sunny day. Similarly, the sunny day irradiance values are highly volatile compared to the cloudy day scenario. Figures 6.4 and 6.5 show the harvested energy in five randomly selected sensor nodes for both scenarios, sunny and cloudy. While the energy harvested in the cloudy scenario was low due to the low irradiation from the sun, the nodes on the sunny scenario depicted notable differences. Some sensor nodes had a stable evolving irradiation exposure and some others show pronounced

peaks during the hours of the day in which solar irradiation is the most elevated. Since sensor nodes are constantly moving, some of them stay within the same region and depicted the same patterns as the collected irradiance values of that region. It is important to note that although some sensors kept their battery fully charged during the times with highest irradiation, it is not guaranteed by all sensors in the simulation. These different behaviors are the results from the heterogeneity in their sun exposure and the functions and activities they carry over.

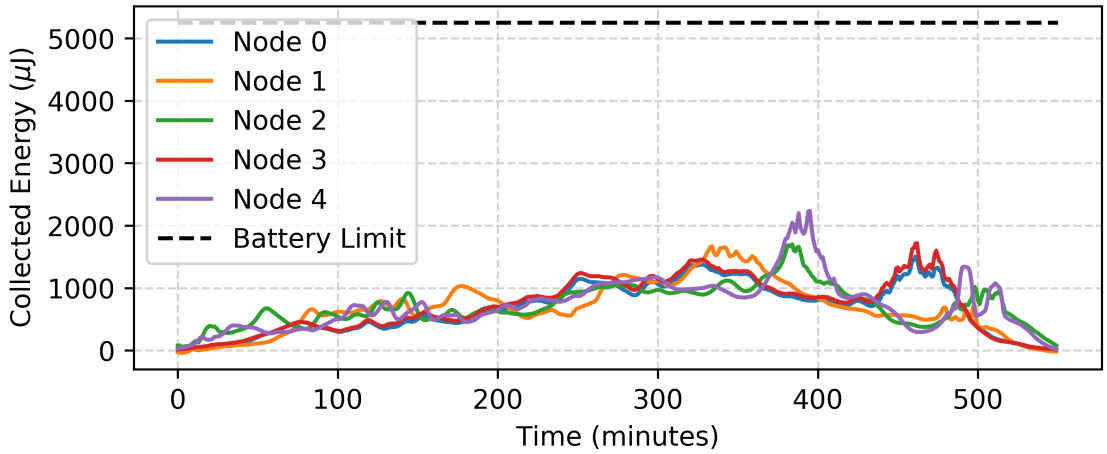


Figure 6.4: Residual energy levels of 5 randomly selected nodes: cloudy 9-hour day

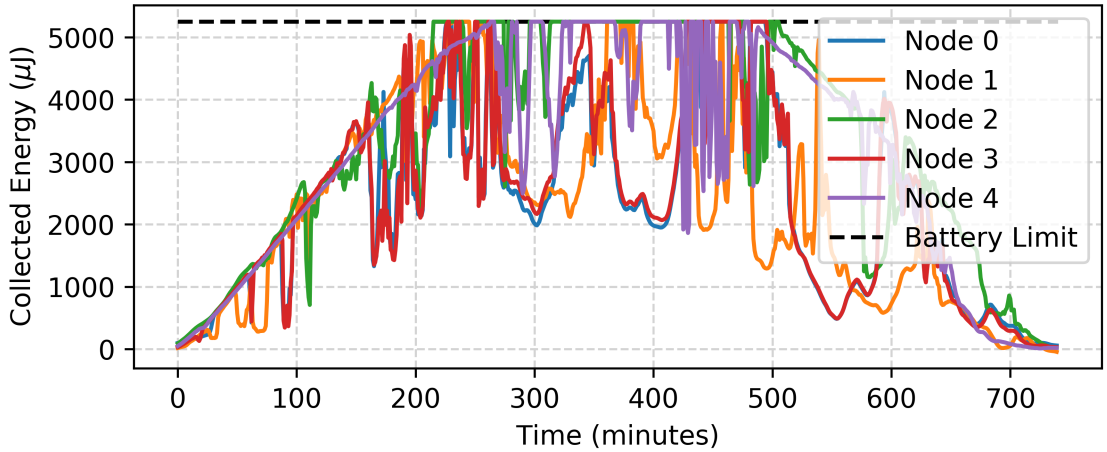


Figure 6.5: Residual energy levels of 5 randomly selected nodes: sunny 12-hour day

In our simulation, 5,500 mobile sensor nodes were placed in an area of 15.86 km^2, where they moved following the Gauss-Markov mobility model [LH99]. This mobility model is time dependent, and it randomly chooses an initial speed and direction. It's next speed and direction, however, depend on the on the previous, so the travel is smoother. In addition, if a selected trip takes the node to the border of our simulation area, it then chooses a direction and speed to take it back to the simulation area when the node finishes its trip. However, other mobility models simply restart the position of nodes, sending them back to the center of the simulation area. Therefore, this mobility model reflects the movement of sensors in real-time applications. The values of the mean speed and the transmission range of each sensor is 20m and 100m, respectively.

We employed a 30-minute time window in which sensor nodes collect data, harvest energy and perform their normal activities. Sensing occurred asynchronously every 30 seconds. The values for threshold ψ varies subject to $\omega = 10\%$. The quantity of missing data samples, which represents the amount of data cleaning workload, are randomly assigned to sensor nodes from 20% to 80% of the total collected sensing values. At the end of the sensing period, sensors computed the onboard energy prediction for the next data cleaning distribution period. Clusters were formed, and sensors forwarded their current and predicted energy together with the number of the dirty data samples contained for the specific time slot. This information was used to perform the data cleaning workload distribution. To evaluate the effectiveness of our results, we compared the count of sensor nodes with energy levels above 10%, the number of dirty data samples not submitted to data cleaning process during its corresponding iteration, and the network welfare when performing data cleaning individually and when performing Leontief-DCD before cleaning the data.

6.4.1 Energy Harvesting

We employ TelosB sensors with variable energy consumption based on two operational modes to characterize the energy harvesting and consumption in EH-MWSN. We considered a scenario where sensor nodes spend 50% of the time in receiv-

ing mode and 50% in transmitting mode, in which sensors consume 24.8mA and 22.8mA, respectively [TEL]. In the same way, we assume that for energy value predicted sensors spend 11.44 mJ and for every data sample cleaned 14.75 mJ. Our sensors are supported by Energizer rechargable batteries with 2000mAh capacity. Our energy-harvesting system is equipped with a solar panel of 0.5 Watts with dimensions 5.5x7cm and 17% efficiency. For this simulation, we initialized our sensor nodes at 20% of its energy storage capacity.

Our simulation is performed in iterations, and the amount of energy that a device can harvest during a time slot t is denoted by $E_{harvest}$ [GWZ11]. We collect the irradiation values in each time slot t of length k, and calculate the energy harvested at every time slot using H_t which is the integral of the irradiance given in J/cm^2 and is computed using the following:

$$H_t = \int_{t=1}^{k} I(t)dt \tag{6.8}$$

where I is the irradiance from the sun in W/cm^2. Additionally, the energy harvested is not only depended on the irradiation during the time slot t, but it depends on the physical characteristics of the energy system of the sensor node and is calculated by:

$$E_{harvest} = (A)(\eta)(H_t) \tag{6.9}$$

where A is the size of the solar panel in cm^2 and η is the efficiency. We then compute energy consumption during the time duration in which the sensor node engages in a specific operation mode:

$$E_{consumption} = (C)(V)(P) \tag{6.10}$$

C is the battery current in Amperes, V is the energy drawn from the operation mode, and P is the duration of the period of time while the sensor was operating in the mode under evaluation. Finally, we compute the onboard residual energy at the

end of time slot t for sensor n using the energy harvested and the energy consumed:

$$E_{(n,t)} = E_{harvest} + E_{(n,t-1)} - E_{consumption} \tag{6.11}$$

6.4.2 Leontief-DCD Results

Leontief-DCD removes the need to trade between sensors' uptime and data quality. It provides a method to centrally plan a strategy to distribute data cleaning workload among cooperative sensors in EH-MWSN, where the mobility of nodes, sun exposure variability and the heterogeneity of sensors' functions challenges the survivability of the network.

Figures 6.6 and 6.7 show the count of sensor nodes that remained active during the cloudy scenario and sunny scenario, respectively. We consider sensor nodes to be active when their onboard residual energy levels are above 10%. In the sunny scenario we can observe that during the highest irradiaiton periods of the day, sensors that did not add the Leontief-DCD to their data cleaning processes depicted similar performance. Nevertheless, Leontief-DCD showed a slight superior performace during the following three hours once the irradiance levels provided by the sun decreased. Moreover, when we observe the cloudy scenario, where irradiation exposure is limited, the implementation of Leontief-DCD increased the sensor count during 6 hours in up to 44.01%. Note that when employing Leontief-DCD, the energy expenditure increased due to the extra computation. It would be expected that an increase in energy expenditure would decrease the sensors' uptime, but because Leontief-DCD distributed the workloads taking in consideration the network as a whole, rather than individual sensor nodes, the extra computation is justified by the increase in sensors' uptime.

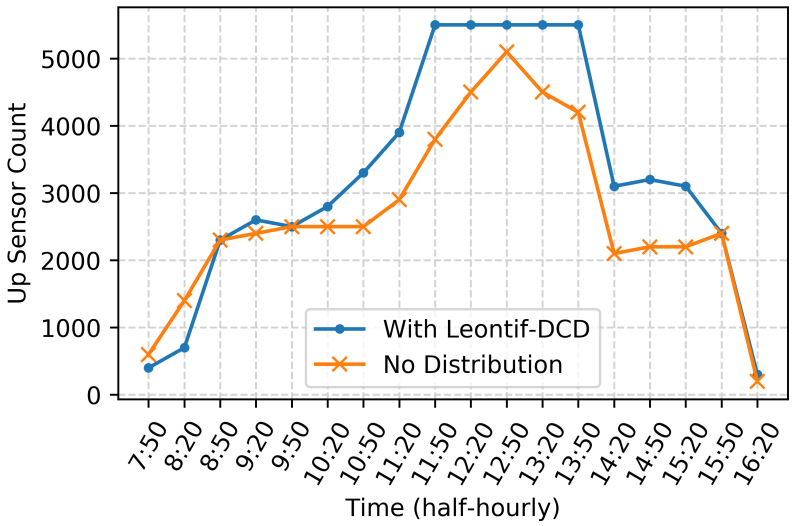


Figure 6.6: Cloudy Scenario: Number of active sensors

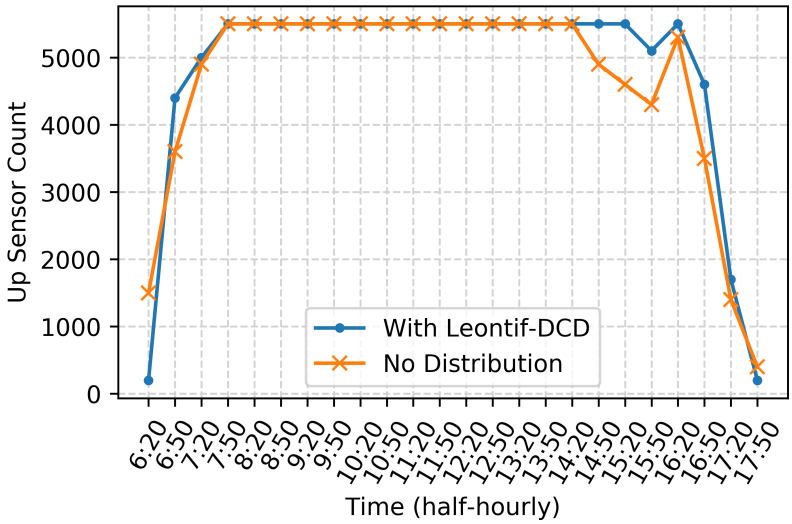


Figure 6.7: Sunny Scenario: Number of active sensors

Figures 6.8 and 6.9 exhibit the number of dirty samples that were not submitted to data cleaning processes during the iteration in which they occurred. In the sunny scenario even when initially there was a high number of dirty samples that were not submitted to data cleaning, in the following 6 hours all dirty samples were submitted to data cleaning in a timely manner in both cases, when sensors

employed Leontief-DCD and when they did not. Notwithstanding, during the last 4 hours when the irradiation patterns tend to decrease, Leontief-DCD increased the number of sensor nodes submitted to data cleaning in up to 25.57%. Moreover, in the cloudy scenario, where the energy of the network is lower, we can see the benefit of employing Leontief-DCD clearly. In this case the number of dirty data samples that were not subject to data cleaning process in its corresponding iteration kept notably lower at all times. Even though the data cleaning process that did not employ Leontief-DCD was able to reduce the number of dirty samples not processed for cleaning during the highest irradiance exposure, it did not process 100% of all dirty data samples at any time. Using Leontief-DCD allowed 100% of dirty data samples to be submitted to data cleaning during the cloudy day simulation, showing its value and usefulness in desicion-making applications. This behavior in the performance of Leontief-DCD shows its ability to increase the possibility to reach a better data quality by distributing the data cleaning workload.

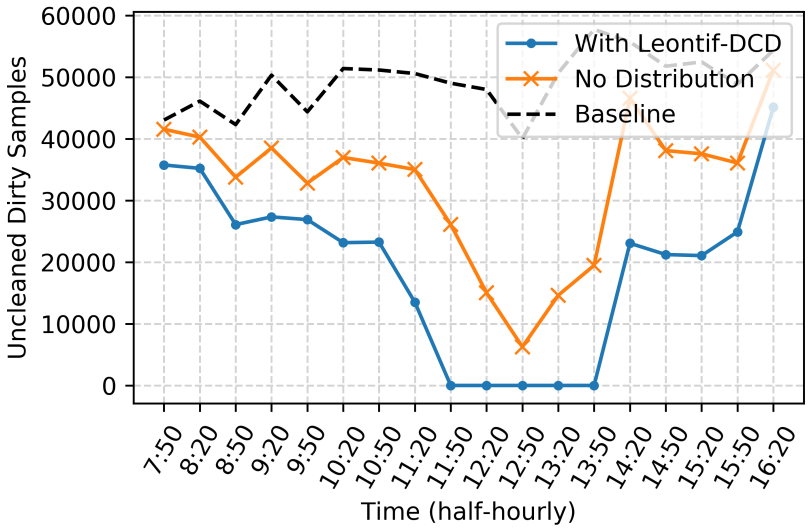


Figure 6.8: Cloudy Scenario: Samples not submitted to data cleaning

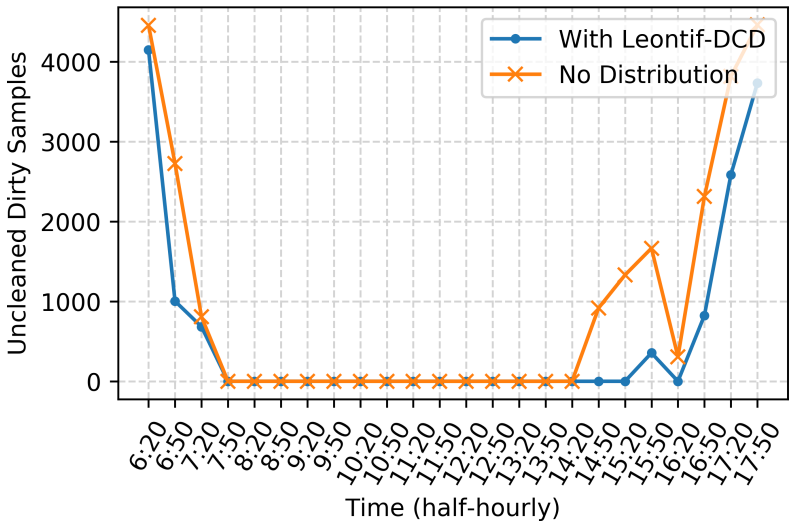


Figure 6.9: Sunny Scenario: Samples not submitted to data cleaning

6.4.3 Network Welfare Analysis

In addition to the performance evaluation shown above, we conducted a network welfare analysis to evaluate the collective well-being of sensors within the network when distributing the data cleaning workload using our Leontief-DCD method. Welfare economics analyzes how the allocation of resources and income distribution affect social welfare. In MWSN, we employ this concept to analyze how the allocation of energy resources by strategically redistributing the data cleaning workload using Leontief-DCD, affects the overall well-being of the network. This network welfare analysis takes into consideration the energy available in the network and the degree of inequality.

To measure the network's well-being, we employ the iso-elastic Social Welfare Function [Atk70], as depicted in Equation 6.12. This iso-elasctic assigns lower welfare values to sensors with already high energy levels and lower values to sensors with low energy levels. In this way, the overall well-being of the network is not dictated by the sensors with the highest energy levels, but by a weighted aggregation

of the network energy distribution. Its functional form is as follows:

$$W = \frac{1}{N}\left(\frac{1}{1-e}\sum_{i=1}^{N}\left[(E_{(i,t)})^{1-e}\right]\right) \tag{6.12}$$

where N is the number of sensor nodes in the network, and $E_{(i,t)}$ is the sensor's onboard residual energy values at the end of each iteration t. Lastly, e is the equality aversion parameter and equals 2/3 for our evaluation.

Figures 6.10 and 6.11 demonstrate the overall well-being of the network. As expected in the sunny scenario, the network welfare depicts comparable values in both cases, when employing Leontief-DCD and when simply implementing data cleaning by itself, due to the high energy availability in the network. On the other hand, in the cloudy day scenario, the results of the network welfare computation when using Leontief-DCD showed a notorious increase when compared to data cleaning processes carried out without distributing the workload as shown in Table 6.2. Leontief-DCD submitted more data to cleaning process in a timely manner, and due to the distribution of workload, the number of sensor count increased and the energy resources in the network were distributed in such a way that the cost of the extra computation got justified by the benefit that Leontief-DCD delivers to the network.

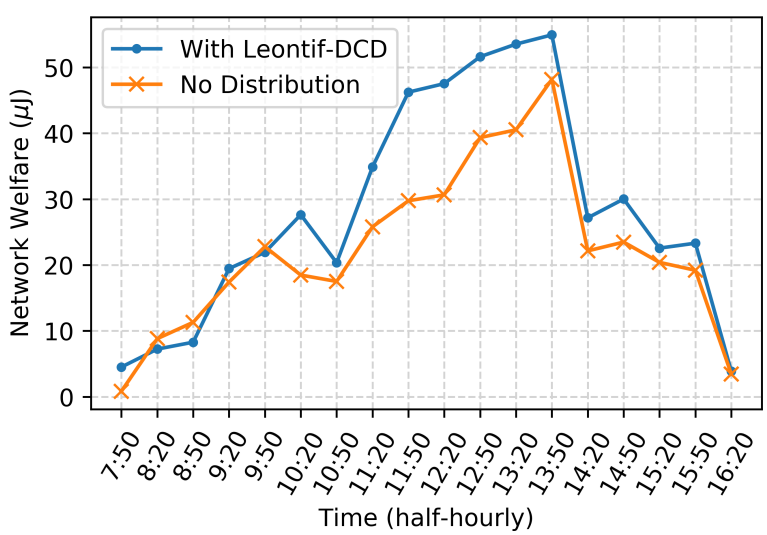


Figure 6.10: Cloudy Scenario: Network Welfare metric

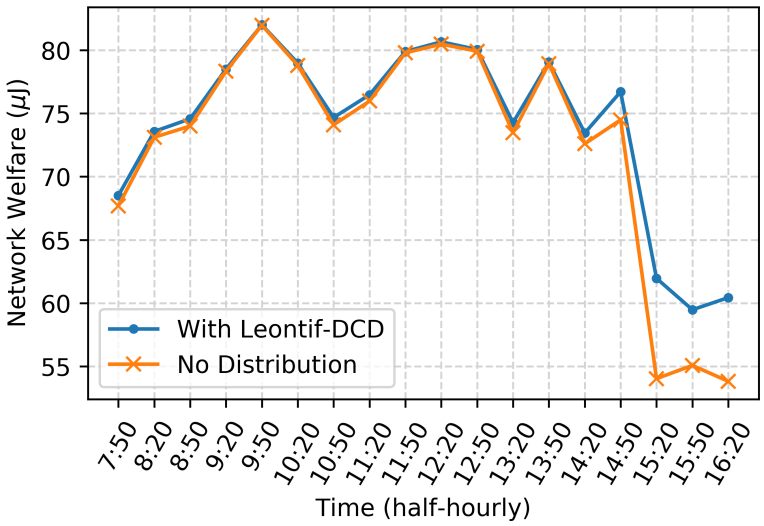


Figure 6.11: Sunny Scenario: Network Welfare metric

Time	Sunny-IDC	Sunny-LDCD	Cloudy-IDC	Cloudy-LDCD
7:50	67.68	68.50	0.83	4.51
8:50	73.99	74.57	11.30	8.28
9:50	81.94	82	22.78	21.92
10:50	74.10	74.68	17.50	20.34
11:50	79.79	79.89	29.75	46.24
12:50	79.92	80.05	39.34	51.62
13:50	78.94	79.06	48.18	54.94
15:50	74.49	76.71	23.48	30.01
16:50	55.07	59.47	19.20	23.31

Table 6.2: Network Welfare per Hour for Individual Data Cleaning and Data Cleaning using Leontief-DCD

6.5 Summary

Leontief-DCD is a novel data cleaning workload distribution method created to employ the onboard residual energy information available at each mobile node to determine whether sensors will be in ENO state in the future. This state determination is accomplished by predicting future onboard residual energy values without any major external information source such as geographical and weather data. The ENO state determination is a crucial information for our Leontief-DCD as it aids in increasing the network uptime. Then, our method uses these predictions to distribute the data cleaning workload based on the Leontief Input-Output closed model, widely utilized in the analysis of global economy production and consumption. Leontief-DCD uses each sensor nodes' energy level information and the amount of dirty data to analyze the energetic input and output and to propose the data cleaning distribution strategy that would drive the network as a whole towards NNO state.

The performance of Leontief-DCD in the face of networks with low energy loads shows its ability to increase the number of dirty samples that are put though data cleaning on time. Additionally, it reduces the sensor quantities unavailable when associated with data cleaning of large volumes of data. We denoted sensors to be unavailable for cleaning when evaluated under 10% of sensors' onboard residual energy. After evaluation under sunny and cloudy conditions, we were able to show that Leontief-DCD is comparable in sunny conditions and provides a significant benefit during cloudy conditions by increasing the number of data samples engaged in data cleaning processes by up to 25.57%, the count of active sensor nodes by up to 44.01%, and the network overall well-being by up to 55.42% compared to when data cleaning was performed by each sensor individually. Lastly, we provided a novel network welfare metric for evaluating the collective performance of a network based on each sensors' onboard energy levels.

To the best of our knowledge this is the first study that involves NNO in EH-MWSN by distributing the data cleaning workload. This combination of approaches collectively contribute to achieve a NNO state without having to compromise any of the sensors' functionality in EH-MWSN dynamic and heterogeneous environments.

CHAPTER 7

LIMITATIONS, FUTURE WORK AND CONCLUSION

In this dissertation, we addressed the challenges that mobile networks encounter in providing reliable data by proposing a set of diverse data handling solutions for MWSN. These mechanisms consider the constraints in sensors' resources and the challenges that mobility adds in producing reliable data. In this chapter, we discuss the contributions and limitations of this research and review the future work and conclusion.

7.1 Limitations

Diversified Trust-Based Data Cleaning for MWSN

We developed and evaluated a data cleaning method that selects a set of sensors to support during the data cleaning process in MWSN. Due to the heavy computation of conventional data cleaning methods, static WSN rely on the presence of a sink and a back-end for their data processing, and since the presence of a sink or A back-end is an unrealistic expectation in mobile scenarios, these methods cannot be extended to MWSN. In our diversified trust-based data cleaning method, we evaluated a set of parameters to measure the trustworthiness of sensor data accuracy based on trajectory behavior similarity and the Spatio-temporal characteristics exhibited in mobile environments. Next, our approach minimizes the error in data estimation by diversifying the risk of trusting the data accuracy of these sensors. Finally, we perform the cleaning by diversifying this risk among the selected set of spatially autocorrelated sensors. This scheme constitutes an effective online system for selecting a trustworthy set of sensors to support during the in-network data cleaning process. By selecting a set of sensors, DTP can find the combination of weights to more accurately predict the values to clean the dirty data. This diversified trust technique reduces the risk involved in trusting a single sensor node's data. DTP demonstrated its outstanding capabilities to consistently achieve high data accuracy, reaching up to 99% of cleaned data with consistent low average percent error, outperforming other approaches.

As limitations of this work, we only considered data received from the source and did not evaluated second-hand data. Because sensor nodes could be within each others' vicinity during their sensing period and may not see each other during the data cleaning period, it can be difficult to find a set of sensors that can provide all the information required to clean data. Moreover, real-world scenarios in MWSN include a combination of the different data loss patterns but, in this work we only considered one type of data loss pattern. Lastly, our method depends on an upper and lower pre-defined boundaries. Sensors with trajectory behavior similarity within these boundaries are employed to support the data cleaning. Nevertheless, if no sensor falls within these boundaries, data cleaning cannot be performed.

Dynamic Trust Weights Allocation to Reconstruct Data in MWSN

To address the limitations of the prior data cleaning method, we designed and tested a technique to reconstruct incomplete data in MWSN. This method considered the different types of data loss patterns that are inherent in MWSN caused by noise and collision, unreliable links, and sensors losing energy or malfunctioning. Moreover, this method considered that the mobility of sensors makes it difficult to find a set of sensors that can provide all the data required to properly execute the cleaning task. Our method is capable of evaluating first and second-hand data and select the most accurate data. It determines the trust level in the data accuracy of each candidate node by evaluating Spatio-temporal correlations, trajectory behavior, quantity and quality of data, and the number of hops traveled by the received data from the source without the use of predefined thresholds. Our results demonstrate that data reconstructed using our dynamic trust allocation method depicts significantly lower Root Mean Square Error (RMSE) compared to methods that only consider Spatio-temporal and sensed values correlations. Our approach showed consistent outstanding performance by achieving high data accuracy when reconstructing sensing data with vast quantities of missing data. The accuracy of this method is obtained from the selection of sensor(s) with the highest quantity of high-quality, spatiotemporally correlated data and with a significant resemblance in trajectory behavior. Our method can be easily tailored to different scenarios in

MWSN, and the flexibility of the modification of weights given to each attribute can contribute to meet specific user requirements. The dynamic adaptive features of this method make it suitable for evaluating the data accuracy for neighboring sensor nodes in scenarios with large quantities of missing data and sensor count, such as in IoT. Contrary to various current methods, our evaluation of trust is not affected by past interactions, which addresses the newcomer problem.

The main limitation of this work includes the assumption that all sensor nodes are collaborative and that the data shared is correct. We did not consider the selfish or malicious behaviors of nodes. Also, sensor nodes are expected to store the data of previous interactions with other nodes to be shared. With the exponential growth of sensing data, it can become a problem for sensors to store the data rather than use what they require and drop the rest.

Data Behavior Modeling and Trend Prediction for Mobile Wireless Sensor Networks in IoT

We developed and assessed the performance of a method to model the behavior of evolving time series data by extracting the features of time partitions and measuring the dissimilarity between consecutive pairs. This dissimilarity measure results in a single value that describes the behavior of data from one time partition to the next and serves as effective input for the SVM to predict the future trend of data. Our method is capable of reducing the space complexity and reach superior prediction accuracy, recall, and precision utilizing only a fraction that represents 5% from the original size of the training data. This data reduction characteristic makes the implementation of BBTP suitable for MWSN, where there is limited memory, low computational capability, and small or irreplaceable power sources. BBTP reduces the amount of data required to be stored and processed in individual sensor nodes. The aftereffect of reducing the size of the data at the node level also corresponds to a reduction of the network traffic, which may lead to fewer message collisions and re-transmissions. Nevertheless, in this method, we did not consider the effects that some contextual parameters may have on the behavior of the data. For example, in a healthcare application, where sensors are tracking the heart rate of a patient,

a sudden spike in heart rate can mean that the patient is about to have a heart attack and an ambulance should be dispatched or that the patient is having a fun ride in the roller coaster at the local fair. The evaluation of contextual parameters can support the identification of the factors causing specific data behavior, which can lead to more accurate data trend predictions.

Data Cleaning Workload Distribution in Energy-Harvesting MWSN

We designed and demonstrated an efficient method for distributing the data cleaning workload in EH-MWSN using Leontief Input-Output model. Our economic-based method sought to benefit the network as a whole rather than individual sensor nodes. In this approach, we proposed the creation of a data cleaning workload distribution strategy that exploits cooperation to drive the network to a state in which for every sensor node, the energy harvested is greater than the energy consumed. This method increases the number of data samples that are run through data cleaning processes and the network uptime. Although the energetic cost rises as the number of data samples ran through data cleaning increases, the network uptime is not reduced. This positive outcome results from the distribution of the overall data cleaning energetic cost, that eases the data cleaning workload in sensors with critically low onboard residual energy. The reduction in the energetic cost experienced by these sensors prevents them from adopting energy-saving strategies that may limit their functionality, including the ability to clean data in a timely manner. It is important to note that our method also improved the overall well-being of the network by up to 55.42% compared to data cleaning performed by each node individually. However, this method assumed that all sensor nodes would agree to collaborate. It did not consider mechanisms to encourage honesty and cooperation from nodes. Also, additional experimentation needs to be made with different real-world datasets.

7.2 Future Work

This dissertation investigated and developed novel techniques to handle data in MWSN. Nevertheless, the increase in applications of MWSN in IoT has incremented the amount of sensing data being generated. Although the value of this data has become one of the most important currencies, future research needs to focus on addressing the challenges that are arising as a result of the increase in sensing data combined with the limitations in resources and the mobility of nodes.

The first part of this dissertation involves cleaning and reconstructing data in MWSN. Sensor nodes in these methods evaluated various parameters to determine trust in data accuracy in a network with symmetric information. Future work can investigate data cleaning and reconstruction methods considering the presence of asymmetric information in the network. This asymmetric information pertains to the fact that sensor nodes where the data originates have more information than the sensor node that receives the data. The uncertainty in the quality of the data received can be studied by employing the "Markets of Lemons" theory. The markets of lemons in MWSN would mean that there are three types of data: good or accurate data, lower quality, or relevant data that can help us to estimate or infer the data we require and lemons or data that was intentionally manipulated. Methodologies that help to overcome a network where only lemons are offered would be a relevant research direction.

Moreover, another part of this research investigates modeling data behavior and predicting data trends in internal sensing for MWSN in IoT. A future research direction points to the utilization of exogenous contextual parameters to determine when data that may look like outliers is data that reflects the true behavior of the phenomena being observed. Another interesting future research direction of this work includes the extension of the use of the data behavior extraction technique to other applications. We consider that this data feature extraction can be employed to detect malicious data manipulation that may pose a threat to MWSN in IoT.

Furthermore, the last section of this work studies the distribution of the data cleaning workload in MWSN. Our future work will focus on incorporating negotiation methods for scenarios containing non-cooperative nodes in EH-MWSN. Utility

functions can be applied to determine the value of the proposed data cleaning strategy for individual sensor nodes. This valuation may help to demonstrate the fairness of the strategy and encourage the collaboration of all sensor nodes. Moreover, negotiation techniques combined with our Leontief-DCD can help in the dynamic redistribution of data cleaning workload based on the negotiated strategy.

Finally, in the future we will focus on implementing the techniques developed in this dissertation in real sensors. This implementation will enable us to conduct various tests to measure and corroborate the performance of the system when executing our methods.

7.3 Conclusion

Mobile wireless sensor networks have become essential elements for modern real-time decision-making applications. These types of applications will transform the way we live. However, the data-centric nature of these applications requires the uninterrupted availability of reliable data to preserve its functionality. This dissertation arises from the recognition of the remarkable importance of resilient data handling mechanisms to prevent applications in MWSN from making erroneous decisions. We investigated and proposed data handling methods that considered the dynamic trajectory behavior relationships among nodes, and the constraints inherent to mobile nodes. This dissertation addressed four main problems when seeking to ensure the availability of reliable and accurate data in mobile environments. First, we developed a method to clean data. Based on the Capital Asset Pricing Model, we evaluated the risk involved in trusting the data accuracy by comparing sensors' trajectory behavior and the Spatio-temporal relationship. We have demonstrated that our method can be used to clean data accurately. Second, we proposed an improvement from our first method to reconstruct highly incomplete sensing data. This method evaluates second-hand and first-hand data accuracy. We showed the ability of this method to accurately reconstruct data with up to 70% of missing data samples without the limitations of boundaries or thresholds. Third, we developed a data behavior modeling method to extract the features of the data that describes its

behavior in a time partition using Japanese Candlesticks and a dissimilarity measure. We showed our method can match the accuracy of other methods while being more efficient in terms of space and training data size. Finally, we proposed a data cleaning workload distribution strategy in EH-MWSN based on the Leontief Input-Output model. We demonstrated that our method favored scenarios with limited energy availability by increasing the data engaged in data cleaning processes and network uptime. To the best of our knowledge, this is one of the first works that apply economic-based principles in mobile and wireless networks. We are optimistic that the outcome from employing economic models in this dissertation research motivates the research community to bridge between economics and problems that remain unsolved in MWSN.

BIBLIOGRAPHY

[AFS17] Tarek Azzabi, Hassene Farhat, and Nabil Sahli. A survey on wsn security issues and military specificities. *2017 International Conference on Advanced Systems and Electric Technologies (ICASET)*, 2017.

[AMAT+18] Kofi Sarpong Adu-Manu, Nadir Adam, Cristiano Tapparello, Hoda Ayatollahi, and Wendi Heinzelman. Energy-harvesting wireless sensor networks. *ACM Transactions on Sensor Networks*, 14(2):1–50, 2018.

[APAK18a] Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, and Georges Kamhoua. A dynamic trust weight allocation technique for data reconstruction in mobile wireless sensor networks. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018.

[APAK18b] Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, and Georges A. Kamhoua. Using candlestick charting and dynamic time warping for data behavior modeling and trend prediction for mwsn in iot. *2018 IEEE International Conference on Big Data (Big Data)*, 2018.

[AQAKS17] Antar Shddad H Abdul-Qawy, Nasr MS Almurisi, A Pradeep Kumar, and T Srinivasulu. Major energy dissipation sources in the iot-based wireless networks. *International Journal of Electronics, Electrical and Computational System*, 6(9):155–161, 2017.

[Atk70] Anthony B Atkinson. On the measurement of inequality. *Journal of Economic Theory*, 2(3):244–263, 1970.

[BH14] Neil W Bergmann and Li-Qun Hou. Energy efficient machine condition monitoring using wsn. In *WCSN*, pages 285–290. IEEE, 2014.

[BM16] Tomáš Bartoň and Petr Musilek. Derivative based prediction with look ahead. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 2118–2123. IEEE, 2016.

[CCH+18] Zhengyu Chen, Lei Chen, Guobing Hu, Wencai Ye, Jin Zhang, and Geng Yang. Data reconstruction in wireless sensor networks from incomplete and erroneous observations. *IEEE Access*, 6:45493–45503, 2018.

[CFSC18] Hongju Cheng, Danyang Feng, Xiaobin Shi, and Chongcheng Chen. Data quality analysis and cleaning strategy for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2018(1), 2018.

[CJKO15] Leszek Chmielewski, Maciej Janowicz, Joanna Kaleta, and Arkadiusz Orłowski. Pattern recognition in the japanese candlesticks. In *Soft computing in computer and information science*, pages 227–234. Springer, 2015.

[Cui18] Sujin Cui. Solar energy prediction and task scheduling for wireless sensor nodes based on long short term memory. *Journal of Physics: Conference Series*, 1074, 2018.

[Dam] Aswath Damodaran. Estimating risk parameters available at http://hdl.handle.net/2451/26789.

[DWW+19] Hong-Ning Dai, Raymond Chi-Wing Wong, Hao Wang, Zibin Zheng, and Athanasios V. Vasilakos. Big data analytics for large-scale wireless networks. *ACM Computing Surveys*, 52(5):1–36, 2019.

[EMZ+15] W. Elghazel, K. Medjaher, N. Zerhouni, J. Bahi, A. Farhat, C. Guyeux, and M. Hakem. Random forests for industrial device functioning diagnostics using wireless sensor networks. 2015.

[Fam66] Eugene F. Fama. *The behavior of stock-market prices.* 1966.

[FD11] Xenofon Fafoutis and Nicola Dragoni. Odmac: An on-demand mac protocol for energy-harvesting-wireless sensor networks. *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks - PE-WASUN 11*, page 49–56, 2011.

[FK17] Abdur Rahim Mohammad Forkan and Ibrahim Khalil. Peace-home: Probabilistic estimation of abnormal clinical events using vital sign correlations for reliable home-based monitoring. *Pervasive and Mobile Computing*, 38:296–311, 2017.

[FZ16] Z. Feng and Y. Zhu. A survey on trajectory data mining: Techniques and applications. *IEEE Access*, 4:2056–2067, 2016.

[GAG+00] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[GASA15] Oussama Ghorbel, Walid Ayedi, Hichem Snoussi, and Mohamed Abid. Fast and efficient outlier detection method in wireless sensor networks. *IEEE Sensors Journal*, 15(6):3403–3411, 2015.

[gen19] *EXCLUSIVE Pentagon's AI Problem Is 'Dirty' Data: Lt. Gen. Shanahan.* Breaking Defense, Nov 2019.

[GHZH14] Yu Gu, Liang He, Ting Zhu, and Tian He. Achieving energy-synchronized communication in energy-harvesting wsn. *ACM Transactions on Embedded Computing Systems*, 13(2s):1–26, 2014.

[GLN15] Saul Gill, Brian Lee, and Euclides Neto. Context aware model-based cleaning of data streams. In *Signals and Systems Conference (ISSC)*, pages 1–6. IEEE, 2015.

[GSB+18] Sam Ghazal, Michael Sauthier, David Brossier, Wassim Bouachir, Philippe Andre Jouvet, and Rita Noumeir. Using machine learning models to predict oxygen saturation following ventilator support adjustment in critically ill children: a single center pilot study. *bioRxiv*, page 334896, 2018.

[GWZ11] Maria Gorlatova, Aya Wallwater, and Gil Zussman. Networking low-power energy harvesting devices: Measurements and algorithms. *2011 Proceedings IEEE INFOCOM*, 2011.

[HAdC+15] David Hasenfratz, Tabita Arn, Ivo de Concini, Olga Saukh, and Lothar Thiele. Health-optimal routing in urban areas. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 398–399. ACM, 2015.

[int] Intel test data available at http://db.csail.mit.edu/labdata/labdata.html.

[IUK16] Md Rashedul Islam, Jia Uddin, and Jong-Myon Kim. Acoustic emission sensor network based fault diagnosis of induction motors using a gabor filter and multiclass support vector machines. *Adhoc & Sensor Wireless Networks*, 34, 2016.

[JAF+06] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. Declarative support for sensor data cleaning. In *International Conference on Pervasive Computing*, pages 83–100. Springer, 2006.

[KAA18] D Praveen Kumar, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu. Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion*, 2018.

[KHZS07] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava. Power management in energy harvesting sensor networks. *ACM Trans. Embed. Comput. Syst.*, 6(4), September 2007.

[KS03] Aman Kansal and Mani Srivastava. An environmental energy harvesting framework for sensor networks. *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED '03.*, pages 481–486, Aug 2003.

[KSY+14] Hisashi Kurasawa, Hiroshi Sato, Atsushi Yamamoto, Hitoshi Kawasaki, Motonori Nakamura, Yohei Fujii, and Hajime Matsumura. Missing sensor value estimation method for participatory sensing environment. In *International Conference on Pervasive Computing and Communications (PerCom)*, pages 103–111. IEEE, 2014.

[KXL+13] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Min-You Wu, and Xue Liu. Data loss and reconstruction in sensor networks. In *International Conference on Computer Communications (INFOCOM)*, pages 1654–1662. IEEE, 2013.

[KXL+14] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Guangshuo Chen, Yu Gu, Min-You Wu, and Xue Liu. Data loss and reconstruction in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 25(11):2818–2828, 2014.

[LBX+16] Jianjun Lei, Haiyang Bi, Ying Xia, Jun Huang, and Haeyoung Bae. An in-network data cleaning approach for wireless sensor networks. *Journal of Intelligent Automation & Soft Computing*, 22(4):599–604, 2016.

[Leo86] Wassily Leontief. *Input output economics.* Oxford University Press, 1986.

[LFS+12] Jason Jingshi Li, Boi Faltings, Olga Saukh, David Hasenfratz, and Jan Beutel. Sensing the air we breathe-the opensense zurich dataset. In *Proceedings of the National Conference on Artificial Intelligence*, volume 1, 2012.

[LH99] B. Liang and Z. J. Haas. Predictive distance-based mobility management for pcs networks. In *IEEE INFOCOM '99.*, volume 3, pages 1377–1384 vol.3, March 1999.

[Ma11] Hua-Dong Ma. Internet of things: Objectives and scientific challenges. *Journal of Computer science and Technology*, 26(6):919–924, 2011.

[Mar52] Harry Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

[mel] Melbourne sensor readings available at https://data.melbourne.vic.gov.au/environment/sensor-readings-with-temperature-light-humidity-ev/ez6b-syvw.

[MHF07] Qiang Miao, Hong-Zhong Huang, and Xianfeng Fan. A comparison study of support vector machines and hidden markov models in machinery condition monitoring. *Journal of Mechanical Science and Technology*, 21(4):607–615, 2007.

[MPV12] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.

[NC04] William Navidi and Tracy Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[Nis01] Steve Nison. *Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East*. Penguin, 2001.

[NP15] Laurent Yamen Njilla and Niki Pissinou. Dynamics of data delivery in mobile ad-hoc networks: A bargaining game approach. *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2015.

[NRE] The national renewable energy laboratory datasets available at https://www.nrel.gov.

[O+18] World Health Organization et al. World health statistics 2018: monitoring health for the sdgs, sustainable development goals. 2018.

[Pea95] Karl Pearson. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[PG16] Raj Kumar Patel and V.K. Giri". Feature selection and classification of mechanical fault of an induction motor using random forest classifier. *Perspectives in Science*, 8:334 – 337, 2016.

[PGWC16] Jaesung Park, Mikhail Gofman, Fan Wu, and Yong-Hoon Choi. Challenges of wireless sensor networks for internet of thing applications. In *International Journal of Distributed Sensor Networks*. SAGE Publications Sage UK: London, England, 2016.

[QWLG18] Zhixin Qi, Hongzhi Wang, Jianzhong Li, and Hong Gao. Impacts of Dirty Data: Experimental Evaluation. *arXiv e-prints*, page arXiv:1803.06071, Mar 2018.

[Res20] Allied Market Research. *Global Sensor Market Opportunities and Forecast 2019-2025*. Mar 2020.

[RMH+19] Mohamad Rida, Abdallah Makhoul, Hassan Harb, David Laiymani, and Mahmoud Barhamgi. Ek-means: A new clustering approach for datasets classification in sensor networks. *Ad Hoc Networks*, 84:158–169, 2019.

[SAPA+18] Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, Kianoosh Boroojeni, Jerry Miller, and Ziqian Ding. Context-aware data cleaning for mobile wireless sensor networks: A diversified trust approach. *International Conference on Computing, Networking and Communications*, 2018.

[SAPA20] Concepcion Sachez Aleman, Niki Pissinou, and Sheila Alemany. *Leontief-Based Data Cleaning Workload Distribution Strategy for EH-MWSN*, pages 1–6, 2020.

[SBB13a] Sukhwinder Sharma, Rakesh Kumar Bansal, and Savina Bansal. Issues and challenges in wireless sensor networks. In *International Conference on Machine Intelligence and Research Advancement (ICMIRA)*, pages 58–62. IEEE, 2013.

[SBB13b] Sukhwinder Sharma, Rakesh Kumar Bansal, and Savina Bansal. Issues and challenges in wireless sensor networks. *2013 International Conference on Machine Intelligence and Research Advancement*, 2013.

[SBS16] S. Samanta, J. N. Bera, and G. Sarkar. Knn based fault diagnosis system for induction motor. *2016 2nd International Conference on Control, Instrumentation, Energy Communication (CIEC)*, pages 304–308, 2016.

[SC07] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[SDBA15] Antesar M Shabut, Keshav P Dahal, Sanat Kumar Bista, and Irfan U Awan. Recommendation based trust model with an effective defence scheme for manets. *IEEE Transactions on Mobile Computing*, 14(10):2101–2115, 2015.

[SGG10] Abhishek B Sharma, Leana Golubchik, and Ramesh Govindan. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, 6(3):23, 2010.

[SJG15] Ruchi Sharma, Gunjan Jain, and ShashiKant Gupta. Enhanced cluster-head selection using round robin technique in wsn. In *International Conference on Communication Networks (ICCN)*, pages 37–42. IEEE, 2015.

[SL09] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[SSCS17] Yuanchao Shu, Kang G. Shin, Jiming Chen, and Youxian Sun. Joint energy replenishment and operation scheduling in wireless rechargeable sensor networks. *IEEE Transactions on Industrial Informatics*, 13(1):125–134, 2017.

[SVR+11] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[SZ16] Faisal Karim Shaikh and Sherali Zeadally. Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 55:1041–1054, 2016.

[TAH+15] Qian Tan, Wei An, Yanni Han, Yanwei Liu, Song Ci, Fang-Ming Shao, and Hui Tang. Energy harvesting aware topology control with power adaptation in wireless sensor networks. *Ad Hoc Networks*, 27:44–56, 2015.

[TCS17] Luca Terruzzi, Riccardo Colombo, and Michele Segata. Poster: On the effects of cooperative platooning on traffic shock waves. *IEEE Vehicular Networking Conference (VNC)*, 2017.

[TEL] Memsic's telosb mote datasheet available at http://www.memsic.comuserfilesfilesdatasheetswsntelosb_datasheet.pdf.

[TNK17] Phan Thanh Noi and Martin Kappas. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors*, 18(1):18, 2017.

[TPI17] Samia Tasnim, Niki Pissinou, and S. S. Iyengar. A novel cleaning approach of environmental sensing data streams. *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017.

[VGB07] Christopher M. Vigorito, Deepak Ganesan, and Andrew G. Barto. Adaptive control of duty cycling in energy-harvesting wireless sensor networks. *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, page 21–30, 2007.

[WTL+17] Jiafu Wan, Shenglong Tang, Di Li, Shiyong Wang, Chengliang Liu, Haider Abbas, and Athanasios V Vasilakos. A manufacturing big data

solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, 13(4):2039–2047, 2017.

[WTX16] Mou Wu, Liansheng Tan, and Naixue Xiong. Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. *Information Sciences*, 329:800–818, 2016.

[WX14] Zhe Wang and Xiangyang Xue. Multi-class support vector machine. In *Support Vector Machines Applications*, pages 23–48. Springer, 2014.

[WY17] Jing Wang and Huili Yue. Food safety pre-warning system based on data mining for a sustainable food supply chain. *Food Control*, 73:223–229, 2017.

[WZWM17] Fei Wang, Zhao Zhen, Bo Wang, and Zengqiang Mi. Comparative study on knn and svm based weather classification models for day ahead short term solar pv power forecasting. *Applied Sciences*, 8(1):28, 2017.

[YH18] Ying-Gao Yue and Ping He. A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions. *Information Fusion*, 44:188–204, 2018.

[YLL+14] Shu Yinbiao, Kang Lee, Peter Lanctot, Fan Jianbin, Hu Hao, Bruce Chow, and Jean-Pierre Desbenoit. Internet of things: wireless sensor networks. *White Paper, International Electrotechnical Commission, http://www. iec. ch*, page 11, 2014.

[YTW+16] Mingyang Yan, Qiuying Tong, Rumin Wang, Changlin Luo, Wuhan Land, Yunbing Gao, and Yuchun Pan. Outliers detection of cultivated land quality grade results based on spatial autocorrelation. *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2016.

[ZCZ+16] Deyu Zhang, Zhigang Chen, Haibo Zhou, Long Chen, and Xuemin (Sherman) Shen. Energy-balanced cooperative transmission based on relay selection and power control in energy harvesting wireless sensor network. *Computer Networks*, 104:189–197, 2016.

[ZSA11] Bo Zhang, Robert Simon, and Hakan Aydin. Maximum utility rate allocation for energy harvesting wireless sensor networks. *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems - MSWiM 11*, page 7–16, 2011.

[ZSS14] Yihong Zhang, Claudia Szabo, and Quan Z Sheng. Cleaning environmental sensing data streams based on individual sensor reliability. In

International Conference on Web Information Systems Engineering, pages 405–414. Springer, 2014.

[ZSS15] Yihong Zhang, Claudia Szabo, and Quan Sheng. An estimation maximization based approach for finding reliable sensors in environmental sensing. In *International Conference on Parallel and Distributed Systems (ICPADS)*, pages 190–197. IEEE, 2015.

VITA

CONCEPCIÓN SÁNCHEZ ALEMÁN

	Born, Panama, Republic of Panama
2009	B.S., Electronics and Communications Engineering Interamerican University of Panama Panama, Republic of Panama
2012	M.S., Telecommunications and Networking Florida International University Miami, Florida
2020	Ph.D., Electrical Engineering Florida International University Miami, Florida

PUBLICATIONS AND PRESENTATIONS

1. Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, Kianoosh Boroojeni, Jerry Miller, Ziqian Ding. *Context-Aware Data Cleaning for Mobile Wireless Sensor Networks: A Diversified Trust Approach.* In 2018 International Conference on Computing, Networking & Communications (ICNC), 2018.

2. Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, and Georges A. Kamhua. *A Dynamic Trust Weight Allocation Technique for Data Reconstruction in Mobile Wireless Sensor Networks.* In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 61-67. IEEE, 2018.

3. Concepcion Sanchez Aleman, Niki Pissinou, Sheila Alemany, and Georges A. Kamhoua. *Using Candlestick Charting and Dynamic Time Warping for Data Behavior Modeling and Trend Prediction for MWSN in IoT.* In 2018 IEEE International Conference on Big Data (Big Data), pp. 2884-2889. IEEE, 2018.

4. Concepcion Sanchez Aleman, Niki Pissinou and Sheila Alemany. *Leontief-Based Data Cleaning Workload Distribution Strategy for EH-MWSN.* In 2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), pp. 1-6. IEEE, 2020. (Best Student Paper Award)