Florida International University

FIU Digital Commons

FIU Electronic Theses and Dissertations | University Graduate School

11-15-2019

# Reputation-aware Trajectory-based Data Mining in the Internet of Things (IoT)

Samia Tasnim
*Florida International University*, stasn002@fiu.edu

Follow this and additional works at: https://digitalcommons.fiu.edu/etd

 Part of the Artificial Intelligence and Robotics Commons, Databases and Information Systems Commons, Digital Communications and Networking Commons, and the Environmental Monitoring Commons

## Recommended Citation

Tasnim, Samia, "Reputation-aware Trajectory-based Data Mining in the Internet of Things (IoT)" (2019). *FIU Electronic Theses and Dissertations*. 4309.
https://digitalcommons.fiu.edu/etd/4309

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

# REPUTATION-AWARE TRAJECTORY-BASED DATA MINING IN THE INTERNET OF THINGS (IOT)

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Samia Tasnim

2019

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

This dissertation, written by Samia Tasnim, and entitled Reputation-aware Trajectory-based Data Mining in the Internet of Things (IoT), 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.

Kang K. Yen

Deng Pan

Leonardo J. Bobadilla

Niki Pissinou, Co-Major Professor

S. S. Iyengar, Co-Major Professor

Date of Defense: November 15, 2019

The dissertation of Samia Tasnim is approved.

Dean John L. Volakis
College of Engineering and Computing

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

Florida International University, 2019

© Copyright 2019 by Samia Tasnim

All rights reserved.

DEDICATION

To my parents and my husband.

ACKNOWLEDGMENTS

This dissertation would not have been possible without the guidance and help of my advisors, Dr. S. S. Iyengar and Dr. Niki Pissinou. In spite of their very busy schedules, they made every effort to guide me throughout the entire process. Their encouraging advice and strict training guided me to conduct independent and ethical research. I am really grateful to them for the tremendous support that I have received, and no word of gratitude is sufficient to convey my appreciation. I am also thankful to my committee members, Dr. Kang Yen, Dr. Deng Pan and Dr. Leonardo Bobadilla for their valuable suggestions regarding my dissertation.

My utmost gratitude goes to my parents, Razia Sultana and Dr. Sheikh Anisuzzaman. With their encouragement, unconditional love and valuable suggestions, I have been able to reach this stage of my academic education. My husband, Dr. Kishwar Ahmed, has accompanied me throughout my doctoral journey. His love, care, faith in me, and support have impelled me through the difficult times of doctoral study. I am grateful to my sister, Sumaiya Tasnim, who motivated me to keep on trying, cheered me up on various occasions. I am also thankful to my cousins Nazia, Naushin, and my parents-in-law for the continuous encouragement throughout my PhD journey.

I would also like to thank the members of Telecommunication and Information Technology Institute ($IT^2$): Shahid, Georges, Concepcion, Dr. Njilla, Hussein, Ricardo and Dr. Guo for giving me much-needed breaks during exhausting research times. I am grateful to Dr. Pumphichet for the initial research discussion. I am thankful to my co-authors Yujian, Juan, Dr. Ahmed, Dr. Chowdhury, Shahid, Dr. Ding and Dr. Boroojeni for the collaboration. Part of the results in chapters 5 and 4 are joint work with Yujian Charles Tang and Juan Caldas, who I mentored as part of the NSF REU program. I also want to extend

my gratitude towards Dr. Jason Liu, Col. Jerry Miller, Olga Carbonell, Rebeca Arocha and Ariana Taglioretti.

This research was supported by the NSF REU site, FIU Graduate school, Air Force Office of Scientific Research and Army Research Office. Thanks to Florida International University for supporting my research through graduate assistance-ship and dissertation year fellowship.

ABSTRACT OF THE DISSERTATION

REPUTATION-AWARE TRAJECTORY-BASED DATA MINING IN THE INTERNET OF THINGS (IOT)

by

Samia Tasnim

Florida International University, 2019

Miami, Florida

Professor S. S. Iyengar, Co-Major Professor

Professor Niki Pissinou, Co-Major Professor

Internet of Things (IoT) is a critically important technology for the acquisition of spatiotemporally dense data in diverse applications, ranging from environmental monitoring to surveillance systems. Such data helps us improve our transportation systems, monitor our air quality and the spread of diseases, respond to natural disasters, and a bevy of other applications. However, IoT sensor data is error-prone due to a number of reasons: sensors may be deployed in hazardous environments, may deplete their energy resources, have mechanical faults, or maybe become the targets of malicious attacks by adversaries. While previous research has attempted to improve the quality of the IoT data, they are limited in terms of better realization of the sensing context and resiliency against malicious attackers in real time. For instance, the data fusion techniques, which process the data in batches, cannot be applied to time-critical applications as they take a long time to respond. Furthermore, context-awareness allows us to examine the sensing environment and react to environmental changes. While previous research has considered geographical context, no related contemporary work has studied how a variety of sensor context (e.g., terrain elevation, wind speed, and user movement during sensing) can be used along with spatiotemporal relationships for online data prediction.

This dissertation aims at developing online methods for data prediction by fusing spatiotemporal and contextual relationships among the participating resource-constrained mobile IoT devices (e.g. smartphones, smart watches, and fitness tracking devices). To achieve this goal, we first introduce a data prediction mechanism that considers the spatiotemporal and contextual relationship among the sensors. Second, we develop a real-time outlier detection approach stemming from a window-based sub-trajectory clustering method for finding behavioral movement similarity in terms of space, time, direction, and location semantics. We relax the prior assumption of cooperative sensors in the concluding section. Finally, we develop a reputation-aware context-based data fusion mechanism by exploiting inter sensor-category correlations. On one hand, this method is capable of defending against false data injection by differentiating malicious and honest participants based on their reported data in real time. On the other hand, this mechanism yields a lower data prediction error rate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

# INTRODUCTION

## 1.1 Background

The rapid development of mobile sensing technologies (e.g., smart devices embedded with various powerful sensors such as temperature, accelerometer, humidity, and gyroscope) has encouraged the proliferation of Internet of Things (IoT). Mainstream smartphones and recently popular wearable devices such as smart watches, fitness tracking devices, and narrative clip are equipped with many sensors. Recently there has been a significant increase in smartphone usage (45.6% increase from 2015 to 2019), making these devices abundant resources of raw data [STA16][HXL+17]. Recent improvements in wireless technology and an increase in smartphone usage, a new mode of data collection (a.k.a. mobile IoT) has emerged [AL17]. Mobile IoT has several practical applications: intelligent transportation systems, air quality monitoring, epidemic disease surveillance, disaster management and environment monitoring [CDW+15], [FZ16], [RGB+17].

HazeWatch [SCHL13], for example, depends on participation by citizens for air pollution monitoring. Agencies such as National Environment Agency of Singapore is now using HazeWatch every day. These applications are typically open to the public and receive sensor data from multiple participants, which influences the reduction of data sparsity at lower costs in comparison with traditional sensor networks. With various advantages, mobile IoT's people-centric architecture contributes to both more inaccurate and corrupted data [MMH+15]. Malicious participants can easily disrupt the IoT data collection process. These entities can interrupt a system by reporting fabricated or erroneous data, making trust evaluation an important

consideration in these applications. Therefore, validating the accuracy of data is essential to ensure the reliability of the application system.

## 1.2 Motivation

In the air quality monitoring application, the sensed pollution data is used to create a pollution map. Because many people rely on the pollution information provided by these applications, imprecise pollution information about an area will mislead people. For example, an asthma patient who prefers a pollution-free route for a walk might be directed to a polluted area to the detriment of his or her physical well-being. In noise monitoring applications, erroneous noise data sensed by participants may result in a wrong noise map for that geographic location. Sick or elderly people who need tranquility may end up in a noisy location due to falsified information. Furthermore, in remote health care monitoring applications, patients carry sensors for glucose monitoring that can communicate with IoT devices and enable real-time medical treatment, known as telemedicine. Without any intelligent data cleaning mechanism, incorrect or missing sensor data received by the healthcare provider will result in wrongful treatment.

The rapid development of mobile sensing technologies (e.g., accelerometer and GPS) embedded in smartphones has dramatically increased the amount of positioning data (also known as trajectories). Discovering various movement relationships (e.g., leader-follower and flock) among trajectories has practical applications in ensuring public safety, transportation, and location-based services. For example, people can be notified immediately about a crime (such as the presence of a stalker) taking place in the vicinity.

Another motivating application is for traffic monitoring: nowadays, Waze [MG13], a driving direction map, is a widely-used crowdsensing application. Drivers receive information from the traffic center as well as from nearby drivers about road and traffic conditions. If the sensing participants report incorrect traffic information, instead of going to a less crowded road, people may be directed to a heavy traffic route wasting time and gasoline. In a more serious scenario, critical information regarding a natural disaster does not reach people due to data error, potentially at a cost of lives. Therefore, we need to detect and correct the data imprecision to improve the data reliability of the applications. Data reliability refers to the condition when data attains enough completeness to be considered for its goal and context [ME14], [MW04].

## 1.3 Research Problem

Existing data prediction techniques, specifically developed for the IoT, focus on using post-processing data cleaning at the server end [ZSS14], [ZSS15], [ZCWL07], [GL15]. These methods cannot ensure real-time data accuracy, as they process data in batches after long intervals of time. Consequently, the methods cannot take immediate action when required. On the other hand, limited research has focused on the geographical context of the sensors [GLN15], [GL15], [LZW+17]. Other contextual information (e.g., wind speed, weather) or semantics have not been considered in state-of-the-art techniques. The dynamism of the experimental environment cannot be exploited while selecting the correlated sensors for data prediction; hence, there does not exist much work in the area of real-time data stream mining.

The problem we are trying to solve in this dissertation is how to enhance the data accuracy in real-time to ensure data reliability in the IoT. We study this problem specifically in terms of missing data prediction, movement behavior-based outlier detection, and data recovery against false data injection attacks for both homogeneous and heterogeneous IoT sensor types. From this perspective, we divide the problem into the following sub-problems: 1) Predict missing data in real time more accurately , 2) Identify movement similarity and detect trajectory outlier in real time, 3) Exploit the correlation between different sensor types to improve the accuracy of the data fusion mechanism, and 4) Predict data more accurately in the presence of malicious participants who inject false data to vandalize the system.

## 1.4 Research Objectives

This dissertation aims to devise novel solutions to ensure robust systems, enhancing data reliability and integrity. Specifically, we investigate the following objectives.

### Context-aware data prediction

Data streams display varied inconsistency and imprecision in mobile IoT applications. The mobility of the sensors causes more data inaccuracy and loss due to short-term network connectivity and data collision [SBB13], [PGWC16]. We hypothesize that spatiotemporal and context correlations will decrease data prediction errors and hence ensure more accurate data stream cleaning. Therefore, our objective is to design and develop a context-aware data prediction model to predict missing data in sensor data streams in a more accurate manner.

### Real-time identification of movement similarity

Data streams demonstrate several unique properties of big data (e.g., volume, velocity, variety, and veracity) and add challenges to data stream mining [CSK+14]. A major challenge is to detect anomalies/outliers in rapid, voluminous streams of data. Our objective is to identify the movement similarity and detect trajectory outliers in real-time.

### Reputation-aware data fusion

Mobile crowdsensing is an emerging sensing paradigm that promotes scalability and reduction in the deployment of specialized sensing devices for large-scale data collection in a decentralized fashion. It also has several practical applications: traffic monitoring, logistics tracking, epidemic disease monitoring, reporting from disaster situations and environment monitoring. Due to its open structure, it allows malicious users to interrupt a system by reporting fabricated or erroneous data, making trust evaluation an important issue in these applications. Our objective is to design and develop a reputation-aware correlated sensor-based data fusion method that is resilient against malicious attackers.

## 1.5 Research Contributions

Our contributions in developing reputation-aware trajectory based data mining methods are summarized as follows.

## Context-aware Data Prediction

While researchers have considered geographical context, no related contemporary work has studied how a variety of sensor context (e.g., terrain elevation, wind speed, user movement during sensing) can be used along with spatiotemporal relationship for online data prediction. We develop online methods by fusing spatiotemporal and context relationships among the participating mobile sensors. To do so, we develop a novel data cleaning mechanism wherein, based on the sensed data and the context relationship of each sensor, we update the credibility of the sensed data [Section 3.3]. Through simulations, we evaluate the performance of our proposed approach, and compare our proposed sensor data stream cleaning method with two state-of-the-art approaches: influence mean cleaning (IMC) [ZSS14] and mean-based cleaning [JAF+06], [SGG10]. Simulation results show up to 24% reduction in root mean square error (RMSE) over IMC and up to 30% over mean-based cleaning [Section 3.4]. Parts of this section has been published in [TPI17].

## Semantic-aware Outlier detection

Semantics is a piece of information that allows us to add meaningful value to the data. For example, GPS coordinates consist of latitude and longitude. The semantic annotation provides these points with a human-readable label such as "restaurant", "school", "bank", etc. Adding semantic annotation facilitates various movement and behavioral patterns identification in mobile trajectory data. In this work, our main contributions are as follows. We propose a method to incorporate geographic domain knowledge to raw trajectory data [Section 4.3]. Additionally, we propose a window-based online sub-trajectory clustering method for finding movement similarity. Our method is able to successfully identify trajectory outliers in the clustering process

with average recall [SL09] 0.92 and F1 score of 0.8 [Section 4.4]. This value of F1 score insinuates the higher classification accuracy of our method. This content was published during my Ph.D. study in [TCP+18].

## Correlated Sensor-based Data Fusion

Smartphones are equipped with different sensors such as humidity, light, accelerometer, and proximity sensor. Amitangshu et al. [PK16] defined a smartphone as a contemporary heterogeneous sensor network. We exploit inter-sensor type correlations while developing our data prediction mechanism, which has the added benefit of being resilient against data corruption attacks. We evaluate our Correlated data and Reputation-Aware data fusion (CDR) method's efficacy in different scenarios based on two datasets: Rome crowdsensing temperature [BBL+14] and Beijing Air quality datasets [ZLH13] [Section 5.4]. The contents of this chapter have been published in [TPI+18].

## Reputation-based Context-aware Data Fusion

We design a reputation-aware data fusion mechanism to ensure data integrity, and develop an online method for data quality prediction in mobile IoT that considers the spatiotemporal, inter-sensor categorization and context relationship among the participants [Section 6.4]. We consider the users who are willing to participate in sensing at the same time. The inter-node distance at a specific instance, as well as user context (e.g., wind speed, sensor model, and user movement during sensing), are considered in correlated sensor selection. We consider differences in spatial granularity while defining the correlated participants for data fusion based on two application types: spatially stable or variable, which was not considered in

the earlier research. Our methodology is resilient against on-off and data corruption attack behavior of a malicious participant. We implement our method on real-world dataset [ZLH13], which we test with the presence of different high numbers (55, 65, 75 and 85 out of 145) of participants injecting false data [Section 6.5]. Our method was able to receive 74% accuracy in the worst case scenario (60% malicious users), exhibiting the quality of resilience [Section 6.5]. In this case, our reputation-based context-aware data fusion (RCoD) outperform the closest competitor reputation system to evaluate participants (RSEP) [AHZ16] by incurring 45.58% less RMSE on average. This research content has been submitted to [TPI+19].

## 1.6 Dissertation Outline

The remainder of this dissertation is organized as follows. We survey the related work in Chapter 2. The context-based data cleaning method is presented in Chapter 3. We focus on addressing the movement similarity identification and trajectory outlier detection in Chapter 4. Then we described the correlated sensor-type based data fusion method and its performance in Chapter 5. In Chapter 6, we present the reputation-based context-aware data fusion mechanism that is resilient against malicious participants. Finally, we present our concluding remarks and provide direction for future work in Chapter 7.

CHAPTER 2

# RELATED WORK

In this section, we discuss the works that are most pertinent to our research. First, we discuss the data mining techniques dealing with static wireless sensor networks (SWSN). With the advent of better wireless technology, mobility of the devices are common in recent days. The works developed for mobile wireless sensor networks are discussed after that. To deal with the trajectory of mobile devices, varied trajectory data mining methods have been invented. We discuss those state-of-the-art techniques in the next subsection. Finally, the different methods developed for mobile IoT data management have been described. We explored the works on data fusion, data reduction, and missing data prediction.

## 2.1 Data Mining in Static Wireless Sensor Networks

Data mining is a crucial part of data management. It is the process used by the user to discover meaningful information in vast data repositories. Appiece et al. [ACFM14] presents the three most popular data mining techniques: predictive modeling, clustering analysis, and anomaly analysis. Data management for wireless sensor networks has been a topic of interest. Data management covers a wide range of issues such as storage, compression, prediction, optimization and data cleaning. Currently, there is a wide range of algorithms that deal with data management. Jeffrey et al. [JAF+06] explained why data cleaning is a crucial step for data management.

In [ZCWL07], cleaning was performed both in sensor level and sink level. Multiple sampling was performed in order to eradicate noise that can be found in data randomly. When the sensor data is regular, few sampling is performed. Their al-

gorithm considered static wireless sensor networks. Participatory sensing is an approach of collecting sensor data using mobile devices. H. Kurasawa et al. [KSY+14] proposed a method to estimate the value of an infrequently used sensor data from frequently used but correlated sensors (e.g., dust limit from temperature, humidity, atmospheric pressure etc.). Along with finding out the missing values, they also appended some data in the training data set for future use by using a locally weighted multiple regression technique. This method performs best when there is a bounty of correlated sensors. Zafeiropoulus et al. [ZSA+09] further discussed the advantages of adding a semantic component to the data, how those semantic annotations enrich the data and ultimately allow for more specific queries. For example, a museum worker would be able to search for the location of the paintings of a specific artist or a specific time period. In contrast, discretization was applied on the continuous numerical values of each sensor to get some categories [IMR14], and inter attribute relationship was calculated by observing the co-appearance of a pair of categories. The data cleaning was performed online by energy-rich entity mobile data collectors (MDC) that move to the polling points of the static wireless sensor networks (WSN) in a periodic fashion. The inter-pole traveling time of MDC was utilized in data cleansing. However, only the spatial relationship among the sensors has been considered, and this method will not work when the sensor nodes are mobile.

In order to detect outliers, both temporal (historical) and spatial correlation among the nodes have been considered [GKD+10] on top of a clustered network structure with two levels of hashing: intra-cluster and inter-cluster. In the first level, a hashing technique based on localization has been used where a cluster head finds out outlier nodes. In the second hashing step, intra-cluster communication takes place where possible outliers can achieve support from measurements performed at nodes belonging to other clusters, resulting in less processing and communication

load for cluster head. However, this method fails to perform when the network structure changes due to the mobility of the sensor nodes. On the other hand, the extended Kalman Filter (EKF) based mechanism was used to predict the future value of neighbors as well as to detect false injected data in online fashion [SSWX13]. A node silently observes its neighbor's transmitted cumulative value and compares it with their own predicted range.

## 2.2 Data Mining in Mobile Wireless Sensor Networks

A context-aware algorithm [SRM+14] was proposed to find out the optimal number of mobility patterns for mobile group formation in decentralized fashion without any prior knowledge about the network. The iterative k-means++ technique was used to find out the optimal group number, and ballot mechanism is used for determining the threshold value. This algorithm does not work in sparse mobile networks.There are some static virtual sensors (VS) located in the center of each square region [PP10]. The term VS insinuates some temporarily allocated memory space at the base station dedicated for storing some value that is being calculated from sensed data of a particular region. To estimate missing VS data, NLMS adaptive filter run prediction model at the base station. In [KXL+13], the authors used compressive sensing to reconstruct missing data. Both temporal stability and spatial correlation feature were used for data reconstruction mechanism.

Pumpichet et al. [PPJP12] developed a centralized method where data cleaning was performed in the base station. In this work, the authors assumed that the trajectory information was accurate, and the data inconsistencies were pre-identified. The authors proposed a method for filling out the previously identified blanks. For each sub-area, a belief table was maintained for the spatiotemporally correlated

mobile sensors. The belief table is global storage of belief measurement for every grid. The belief values were used as a weight in the missing value prediction method. However, the sensor context was not included in this paper. In contrast, the sketch-based data cleaning method [PJP13] was developed for applications (e.g., blood pressure and heart rate monitoring) where inter-sensor correlation did not work. Thus, a method similar to peer to peer was proposed for cleaning data streams that is not dense. A mobile sensor and the base station were the peers in this scenario. However, this method required a high storage space. Also, due to a high rate of data transmission, the power consumption was high for such a sketch-based method.

## 2.3 Trajectory Data Mining

Zhixian Yan et al. [YS+09] presented a multi-stage model to cope with the complexity of trajectory semantics. The multi-stage model took raw movement data and created raw trajectories which were later converted to structured trajectories. Mingqi et al. [LCC12] presented a framework that discovers the users' semantic places from their GPS data. This framework included a multi-layered model that extracts physical places by using a density-based algorithm.

There are some works on the clustering of mobile objects based on the whole trajectory of the movement nodes [NP06, PKK+11, VKG02]. In these cases, the whole lifetime of the trajectories was assessed for discovering cluster. In [LHW07], the authors considered sub-trajectory based clustering of trajectories considering only spatial relationship. However, since the inclusion of time was missing, it was not capable of finding out the exact relationship between various mobile trajectories. Spatial clustering was used for moving object activity discovery. Silva et al. [dSZdM16] used sub-trajectory based clustering considering space and time. Though, they did

not include semantics in their trajectory clustering method. Palma et al. [PBKA08] used a density-based clustering algorithm to find important places from the trajectory. There is also research on shape-based clustering that looks into the shape of the trajectory for similarity finding. A shape-based similarity query was defined in [YAS03]. In contrast, there is significant work based on time-dependent clustering. In this type of works, periodic patterns among movement trajectories are analyzed [NP06, YP12]. Ying et al. [YCL+14] proposed a prediction model based on cluster-based prediction strategy to find out the probable location that a user may move towards based on a similar user's semantic trajectory analysis.

## 2.4 Data Mining in the Internet of Things (IoT)

Chen et al. [CDL+19] proposed a deep learning-based model for urban air quality monitoring. Even though the authors were able to predict and forecast air quality index values exploiting spatial as well as temporal models, it required higher processing time and capability. Also, it is completely supervised, cannot work on unlabeled data. On the contrary, a multi linear regression model was used for forecasting air pollution index [GYT10]. Kumar et al. [KJ10] utilized an autoregressive integrated moving average model for predicting the air pollutant concentrations. Furthermore, Cheng et al [CHZT19] focused on calibration error reduction in air quality monitoring sensors by utilizing spatial correlation and multi-sensor fusion. However, network structure was static, i.e., the densely deployed air quality monitoring sensors do not change their position.

Nowadays, instead of traditional static wireless sensor networks, sensing is distributed among a crowd of people. This brings heterogeneity in the sensor networks and makes the computation more complex. The most recent work on data quality

estimation in mobile crowdsensing is done by Shengzhong et al. [LZW+17]. The authors introduced a context-aware method for data quality estimation in real time. The limitation of this work is that the authors considered the presence of exactly one mobile user at each point of interest (PoI). Kishino et al. [KTS+17] mounted sensor nodes on garbage trucks that drive around the city. Their motivation was to detect target events by analyzing vehicle-mounted sensor data streams. The authors used machine learning methods to detect the target events. On the other hand, the author [Kou18] broached a new sampling method named stratified sampling for calculating the mean temperature of a linear area. In this article, only the random waypoint mobility model was considered for the movement of the sensing devices.

Vitello et al. [VCF+18a] built a simulator for large scale mobile crowdsensing. In this paper, the authors applied a procedure to augment the precision of the graph describing the street network provided by OpenStreetMap (OSM). They could not directly use OSM as dead ends as the intersections are included in the map. They applied some algorithm to adjust the map input while generating large scale mobile crowdsensing participants' trajectories imitating real-world pedestrian movements. Furthermore, Villeto et al. [VCF+18b] proposed an energy-efficient collaborative data collection and delivery mechanism well suited for smart city applications. Based on the proximity of the mobile devices, groups are formed who can communicate through Wi-Fi rather than 3G/4G communication. As a result, the phone's battery is saved. One of the members is elected as group owner who forwards the collected group data to the data collector. However, the efficiency of the proposed method lies in the proper selection of the owner. The authors proposed three policies compliant with current Android implementation of Wi-Fi Direct for this purpose. In the first policy, each grid is termed as a group and, in the second policy, users located around a point of interest are grouped together.

The third policy declares pedestrians walking in the same direction along the street as a group. By using their own developed CrowdSenSim simulator, Villeto et al. showed that the collaborative method outperforms individual sensing methods. Due to not considering the geographic context of the participants, the dynamism of the evaluation environment has not been taken into account in the group formation for collaborative sensing.

## Different Real-world Applications

• Creekwatch [KRZ+11], a smartphone application, allows the monitoring of the conditions of watershed using crowdsensed data. It was invented by the IBM Almaden research center.

• An environmental air quality sensing system was deployed on street sweeping vehicles to monitor air quality in San Francisco [AHM+09].

• Garbage Watch [RS11] employs citizens to monitor the content of recycling bins to enhance the recycling program.

The research group from the University of Luxemburg developed one of the first mobile crowdsensing simulators named CrowdSenSim [FCC+17]. It allows researchers to perform simulations over urban environments. It has been successfully applied in the performance evaluation of city-wide public street lighting [CFK+17a], energy efficient data collection [CFKB17] and user recruitment in crowdsensing campaigns [CFK+17b].

On the contrary, Chessa et al. [CGF+17] proposed a methodology capturing urban morphology with arbitrary precision. It is an axial line method, not detailed enough to capture the street map since the streets are not defined in terms of latitude, longitude pairs.

### 2.4.1 Data Fusion

There exist many approaches to sensor data fusion. In this section, we discuss some of the major contributions.

Analyzing data obtained from Smart Santander, [JGB14] have proposed sensor data fusion technique to comprehend behavioral pattern of humans. Applying poison model, the method in the paper [JGB14] correlated traffic flows and temperature. On the other hand, project [SBB+15] estimated the allurement of smart cities to its tourists. The work collects large number of data in three categories (e.g., photos, tweets, and card transactions by tourists) and fuse sensor data to calculate attractiveness of cities. A number of interesting conclusions were derived in the project [SBB+15], such as contributing factors to tourists attraction to a particular city, etc.

In the paper [AAB+14], authors presented a big data platform to collect, aggregate and visually present smart city data flow. The work focused on identifying city scale events (e.g., event duration and number of visitors) through sensor data fusion and used various types of data sources (e.g., social media, traffic flow). OpenIoT [SKH+15] is an IoT platform that provides semantic services in the cloud. It provides common model platforms for representing sensors (virtual and physical) and instigates various IoT protocols (e.g., CoAP, 6LoWPAN). Most importantly, OpenIoT supports mobile sensors. Furthermore, The Padova project developed by Zanella et al. [CZVZ14, ZBC+14] uses different network layer protocols (e.g., IPv4, IPv6) and collects WS data through the sensor network infrastructure. A survey of different architecture, techniques, and protocols for urban IoT is also presented in the paper [ZBC+14] by Zanella et al. Smart Santander is a European smart city project. The paper [TMC13] reports the various challenges and findings from the Smart Santander project, including sensor data fusion employment at various smart

city places (e.g., parking, home garden). Moreover, An urban information system to build a smart city is presented by Lin et al. in the paper [JGMP14]. The paper uses a noise mapping scenario to show the architecture.

### 2.4.2 Data Reduction

There are various works focusing on the cleaning of data streams. Most of the previous works on sensor data cleaning focused on the reduction of consumed energy. To achieve this reduction, the authors [BGS+13], [DKV+09], [GKD+10] tried to reduce the inter-node communication. In these works, it was assumed that sensor data are always aggregated during submission. There have been significant works on using compressive sensing for data reconstruction in static sensor networks [CLK+13], [KXL+13].

In recent days, researchers [GLN15], [KWW+16], [LJS+17], [LTTH16], [LZW+17], [TCP+18] are designing frameworks to deal with big data services. In the past, the data size was not as big as present days, which influences researchers to design and develop scalable mechanisms to correct any kind of inaccuracy in data streams. For instance, Liu et al. [LTTH16] designed a framework for big data cleaning. This paper gives direction on how to achieve a reliable database in big data applications. They used context to find similarity between data items. Moreover, the authors exploited the usage pattern to classify and group data items that are not related contextually. One of the daunting tasks in dealing with big data is to shrink the data size by extracting the irrelevant subset. Dong et al. [DSS12], in contrast, debated that having more data does not always provide more information. During data integration, proper selection of reliable source among all available sources results in higher data accuracy.

### 2.4.3 Missing Data Prediction

Zhang et al. [ZSS14] proposed data cleaning method for environmental sensing. Depending on the sensed value, the authors tuned each sensor's reliability value on an incremental basis. On the contrary, Peng et al. [PWC15] used unsupervised learning for data quality estimation. This method works after the collection of historical data from all the users; hence it is not an online method.

Trustworthiness was considered as a measure of data quality estimation [LMB10, MMH+15, YSS18]. However, Mousa et al. [MMH+15] used synthetic data set for the experimentation. Huang et al. [HKH14] showed that using a reputation framework helped to weed out non-colluding malicious attackers. Their reputation framework produced more accurate results than not using a reputation framework. However, the authors assumed that data is coming from every discrete block of space-time which is not practical in real-world scenarios. Moreover, Alswailim et al. [AHZ16] proposed a method named Reputation System to Evaluate Participants (RSEP) to cluster participants into three groups based on the sensed data. If the data was within 10% error of the ground truth, it was considered as correct. The winner group was given reward in terms of increased reputation, and the reputation value of members belonging to the remaining two groups were reduced. However, the maximum error that RSEP could identify was only 30%. Also, they assumed that ground truth data is known a priori, which is not practical in real-life applications.

CHMM has been used in the area of computer vision [YFW15],[PKL12], wavelet domain [CB97] and image recognition [ZWL07]. Target appearance change during tracking is always a challenging problem for tracking visual objects [YFW15]. CHMM was also used in dynamic behavior analysis of power distribution networks, equipped with phasor measurement units (PMU), with the aim of providing adequate assistance to diagnostic and control application [ZAS16]. Moreover, Bushra

et al. [AS16] incorporated autoregressive hidden Markov model to detect malicious nodes in static wireless sensor networks. Since it is an autoregressive model, it required high processing capabilities. This method fails to identify malicious nodes when the nodes move.

Another aspect of literature focuses on finding outliers in sensor data streams. In order to find global outliers in the data, Branch et al. [BGS+13] proposed a distance based ranking method. The other existing methods for finding outliers in sensor data are geometry-based [BD12], polygon-based spatial outlier detection [FG09], clustering-based [KZX10], kernel density-based [SPP+06] and histogram approach [SLMJ07]. Bosman et al. [BIT+17] tried to answer the question if adding more neighbors makes the anomaly detection perform better. This paper considered static sensor nodes and it varied the neighborhood size by changing the communication range of the sensors.

CHAPTER 3

# CONTEXT-AWARE DATA CLEANING IN MOBILE WIRELESS SENSOR NETWORKS

In mobile Wireless Sensor Networks (mWSN), uncertainty is a common phenomenon where nodes change their positions rapidly and unpredictably. Although reliability and accuracy are of utmost importance in many sensor applications, it is often difficult to ensure these properties. For example, energy scarcity, frequent movement of the sensor nodes and instability of the communication channel often contribute to the imprecise or dirty data [Eln03, JAF+06]. To recover the lost data, researchers need to find out the correlated sensor data that can be used in the prediction method. In this chapter, we present our novel context-aware method for cleaning mobile wireless sensor networks data streams.

We organize this chapter as follows. In section 3.1, we discuss the background information and challenges. In section 3.2, we discuss the problem and our hypothesis. In section 3.3, we describe different modules of our overall system and present algorithm for efficient data cleaning. Next, in section 3.4, first, we discuss the simulation setup, and then we present our results and analysis. Finally, discussion and summary are offered in section 3.5.

## 3.1 Introduction

Data cleaning deals with missing values, noisy data, inconsistent data, etc. [HPK11]. Recent research works have focused on using post-processing data cleaning at the server end. There also exist some efforts on using online data stream cleaning method in mWSN [PP10]. In static sensor networks, it is sufficient to consider the temporal relationship [EN03, PS07, MB12]. In many previous works, spatiotemporal

relationship among the sensor nodes have been considered to select the candidate sensor for data cleaning [KSY+14, JAF+05, JAF+06, WLT10]. However, not much work has focused on how sensor context can be used in sensor selection for data cleaning.

In environmental sensing, different sensors are deployed to sense various environmental properties (e.g., humidity, temperature, and ozone ) [JJLB12] The sensors are mounted on top of different vehicles (e.g., bicycle, bus, private car, and tram) or carried by a human being, which change their positions very frequently and unpredictably. Therefore, the network structure for these sensor nodes changes dynamically; making it imperative to consider the mobility pattern for sensor data. In some cases, real-time significant decision is made based on the sensed value which raises the necessity of data cleaning. In our work, we focus on both context and mobility of the sensor.

Zhang et al. [ZSS14] proposed data cleaning method for environmental sensing. Depending on the sensed value, the authors tuned each sensor's reliability value on an incremental basis. The reliability measurement does not require prior hardware knowledge. With the advance of time, they incrementally adjusted the reliability of each sensor based on their sensing data accuracy. In many cases, if only the spatial and temporal relationship among the sensing nodes are considered, then the data inaccuracy is not identified. For example, a pair of sensors may be located in a nearby location. The GPS co-ordinate (longitude, latitude) value might be quite close. If only spatial and temporal relationship is incorporated, these pair is an ideal candidate to be declared co-related sensors. However, there is a thin wall separating these two nodes. Due to this wall, one node is at indoor while the other is placed at outdoor. As a result, there is a high possibility that the sensed value will vary for this pair of nodes. If data from one of this pair is used to clean the missing value of

the other, the accuracy will degrade. Given the mentioned limitation, we propose an algorithm to efficiently calculate the credibility of each sensor during the execution period. In our method, we considered not only the sensed data value but also the context of the sensor along with the mobility pattern of the mobile sensors. With the combination of both of these comparisons, our method performs better cleaning and helps in ensuring higher accuracy in environmental sensing.

## 3.2 Problem Statement

Data streams display varied inconsistency and imprecision in mobile IoT applications. The mobility of the sensors causes more data inaccuracy and loss due to short-term network connectivity and data collision. Thus, it is needed to correct those data imprecision for ensuring the data reliability. The research question that we want to answer is: how can the missing data in sensor data streams be predicted in a more accurate manner in mobile sensor networks?

Existing data prediction techniques, specifically developed for the IoT, focus on using post-processing data cleaning at the server end. These methods cannot ensure real-time data accuracy, as they process data in batches after long intervals of time. Consequently, the methods cannot take immediate action when required. Our hypothesis is that spatiotemporal and context correlations will decrease data prediction errors and hence ensure more accurate data stream cleaning. We include contextual correlation along with spatiotemporal correlation. This is needed to take care of the dynamism of the experimental environment caused by the mobility of the sensor nodes.

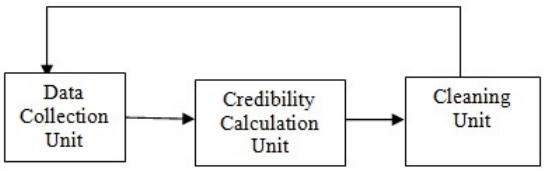



Figure 3.1: Overall Architecture Design

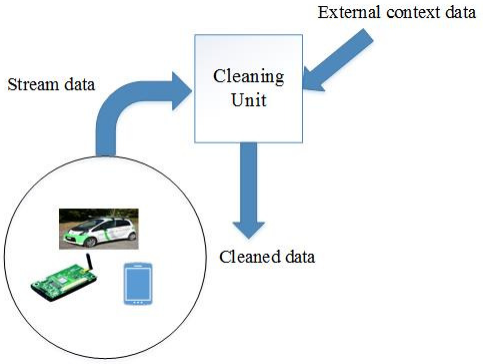



Figure 3.2: Cleaning Unit

## 3.3 Architecture

In our system, we calculate sensor credibility according to the historical sensing performance as well as considering the context of the sensor. Data credibility or alternatively reliability is defined as a state that exists when data attains enough completeness and error-free to be considered for its goal and context [ME14], [MW04]. We have used a sliding window based weighted moving average mechanism. By context, we mean the environmental space, time, physical condition around the sensor, the carrier of the sensor etc. We have considered Dynamic data-driven application method where there is an interaction between the prediction model and data acquisition. Fig. 3.1 depicts the feedback loop from the cleaning unit towards the data collection unit. The credibility calculation module evaluates different weight values corresponding to sensed data, sensor context and mobility pattern of the sen-

sor nodes. The cleaning unit has been shown in Fig. 3.2. The data streams from heterogeneous mobile devices are input to the cleaning unit. External context data is added before performing the cleaning. In the end, the cleaned data is produced from this unit.

### Sensor Credibility Module

Sensor credibility is the ratio of the number of times a sensor senses data correctly and the total number of sensing performed by that sensor during the desired time period [ZSS14]. The weight is dependent on the closeness of the sensed data and predicted value. For all the data samples that a sensor senses during a particular time window, the difference is compared with the predicted value. The mean value of all data samples from spatially correlated sensors is calculated and used as the predicted value. If the difference is in a tolerable range, the sensed data is considered correct. The number of correct is divided by the number of samples to attain the credibility (or reliability) of the sensor. The sensor with higher credibility has a higher impact during correlation.

### Context Credibility Module

This module takes care of any change in the context of the mobile sensor. Context is a vector of multiple properties of the sensor (table 3.1), that is taking part in physical environment sensing. In mobile environments, sensor context is of high importance. In static sensor networks, sensor context alters rarely. Due to mobility and heterogeneity of the nodes, there is a high probability that sensor context may vary as time proceeds.

Table 3.1: Factors considered for data cleaning

| | |
|---|---|
| External Factor | Dust, Snow |
| Internal Factor | Battery level, hardware malfunction |
| Context | Terrain elevation, wind speed, uneven road |
| Mobility Pattern | Random Waypoint, Nomadic |

We calculate context credibility by comparing the context value of the sensor with the neighboring sensors co-located during a similar time window. If the difference is within a tolerable limit, then the higher weight is being assigned. We have added context information (e.g., land elevation, population density) by using API [War13]. By incorporating context, we bring dynamism in area classification. Two nodes located nearby in the same square region attain different levels of importance due to their context value (e.g., change in terrain elevation, height above sea level).

## Mobility Pattern-based Credibility Module

We need this module due to the mobility of the sensor nodes. We assume that every trajectory has been recorded in an environment with good position accuracy, and therefore the location information is correct.

We have analyzed the movement patterns of the sensor nodes. Higher weight is being assigned to the sensor nodes that tend to move in close correspondence with the sensor that needs data cleaning during the desired time window. Sensors that move in a group are given more importance. In the random waypoint model, nodes move independently to a randomly chosen destination with a randomly selected velocity. It includes pause times between changes in direction and/or speed [CBD02]. In the nomadic mobility model, a group of nodes moves from one place to another [CBD02].

**Algorithm 1** Cleaning Algorithm

**Input :** sensor_id

1: block = region[sensor_id]
2: context_val ← context[*sensor_id*]
3: $S \leftarrow$ for each $s_i \in$ block during $t_w$
4: **for all** $S_i \in S$ **do**
5:  **for all** $r_i \in S_i$ **do**
6:  $w_1 \leftarrow$ Equation (3.2)
7:  $w_2 \leftarrow$ CalcContextwt($S_i$,context_val)
8:  $w_3 \leftarrow$ MobilityAffinity($S_i$,sensor_id)
9:  **end for**
10:  sum $\leftarrow r_i * w_1 * w_2 * w_3$
11:  weight $\leftarrow w_1 * w_2 * w_3$
12:  totalSum $\leftarrow$ totalSum + sum
13:  totalWeight $\leftarrow$ totalWeight + weight
14: **end for**
15: cleanedData $\leftarrow \dfrac{\text{totalSum}}{\text{totalWeight}}$

## Data Cleaning Module

This module performs the cleaning of dirty data. We have designed a centralized online data cleaning mechanism. We assume that there is a pre-processing mechanism that identifies the values that need to be cleaned. Our novel algorithm predicts the replacement value. To do so, we considered different correlations to find out the desired sensor data to be used for data stream cleaning.

$$Pred = \frac{1}{n} \sum R. \tag{3.1}$$

Now, we explain our cleaning algorithm (given in Algorithm 1). Each sensor samples the sensed value and sends the sensed data along with their location to the central unit. For example, sensor node i samples the sensed values, $r_i t_{11}$ , $r_i t_{12}$ , ..., $r_i t_{1m}$, in the first time window $[T_1^{(s)}$ , $T_1^{(f)}$ ]. Similarly, all sensors sense and transmit their data during the whole time cycle. In order to predict the missing or corrupted data value of a particular sensor, first, the data credibility value of the

sensors co-located at the region of that sensor during that time window is considered. During each time window, the mean value of all readings made by sensors located in the square region is calculated according to Eqn. 3.1. This mean value is used as the predicted value while calculating data credibility. If the difference between a sensed value (e.g., $r_i t_{11}$) and the predicted value is within the threshold limit, we assign higher credibility for this sensed value. On the other hand, if the difference is greater, that sensed value is not considered for the cleaning of the dirty data according to Eqn. 3.2.

$$cred(data) = \begin{cases} 1, & |data - predicted| \leq threshold_d, \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

To calculate the context credibility, we compare the context of spatially co-located nodes and the node that needs data replacement. This procedure is called `CalcContextwt` in Algorithm 1.

The procedure `MobilityAffinity` evaluates the mobility relationship between two sensor nodes. The sensor id of the node requiring data cleaning and the sensor id of the co-located node is input to this procedure. If this pair of nodes show nomadic movement behavior, $w_{mob}$ obtains a higher weight to indicate high mobility affinity between them.

For all data of a sensor, the three weight values $(w_1, w_2, w_3)$ are calculated by calling Eqn. 3.2, `CalcContextwt` and `MobilityAffinity` respectively. We calculate the product of these three weight values and assign it to $weight$. These three weight values control the significance of each sensed data value in the prediction of the clean data. The product of the weights and sensed data $(r_i)$ is stored in $sum$. $totalSum$ stores cumulative sum value for all sensors and $totalWeight$ stores

cumulative weight. These variables are used for the weighted moving average based *cleanedData* calculation.

## 3.4 Performance Evaluation

### 3.4.1 Simulation Setting

We simulated a scenario of 10 mobile sensors moving in an area of $200m * 200m$ using ns-2 [KF11] and BonnMotion [AEGPS10]. The simulation duration was $1000s$. Some of the mobile nodes moved in nomadic mobility pattern, and others followed random waypoint mobility. Our proposed algorithm (along with IMC and Mean-based cleaning) was tested on a dataset of Smart City project in Melbourne [CoM18]. Environmental data consisted of humidity and temperature values. Terrain elevation was added as context information using API [War13].

We used faulty data injection mechanism to introduce an impurity into the sensed data. We applied an intense spike similar to Fig. 3.3 on the data to introduce error. Usually, hardware malfunctioning or battery exhaustion causes this type of intense spike [NRC+09]. When the spike is applied, each data has a 50% chance of replaced by spike value.

During each iteration, sensor nodes sensed the humidity of the region it is located. The whole region is divided into 100 equally sized blocks each having dimension of $20m * 20m$. The readings having a spatial position within the same block were considered as a singular group. We considered fixed and variable speed for the mobile nodes. For the case of walking, the speed value was set to 2 mph. On the other hand, we used 30 mph speed value for the case when the mobile sensor is assumed to be attached over a slow-moving car.

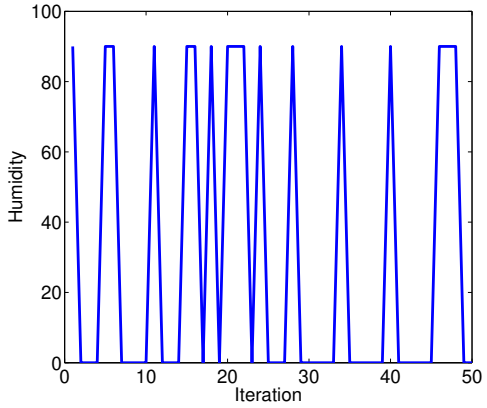



Figure 3.3: Spike Data.

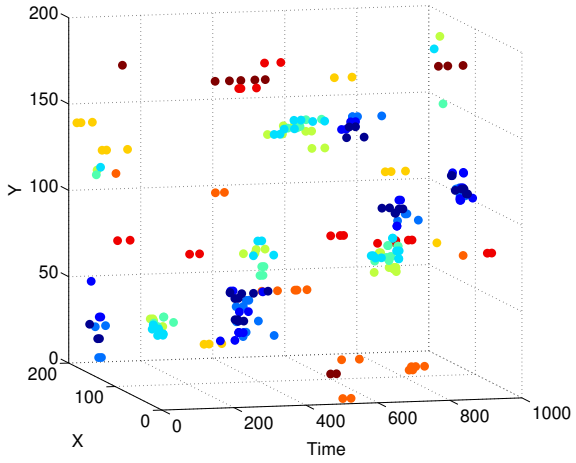



Figure 3.4: Positioning of different nodes.

### 3.4.2 Results and Analysis

We calculated RMSE and used it as a performance measurement criteria of our algorithm. RMSE is a standard metric to evaluate the accuracy of the prediction model [ZSS14].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{V}_i - V_i)^2}, \tag{3.3}$$

where $\widehat{V}_i$ is the predicted value and $V_i$ is the original value. The Mean Square Error (MSE) is the average of squared deviations of the predictions from the true values. RMSE is calculated by finding the square root of MSE. Since we knew the true values from the original dataset before applying the spike value, we could calculate the RMSE value from the predicted values.

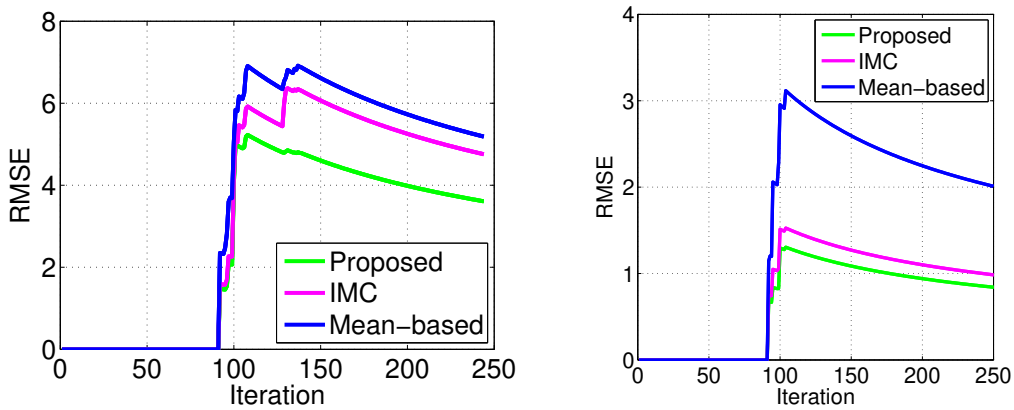



Figure 3.5: RMSE comparison for (a) continuous walking of the nodes, and (b) variable movement speed.

Fig. 3.5 (a) depicts the case of all the mobile nodes moving continuously in walking speed (e.g., 2mph) and sensing the humidity values. A sudden rise in the RMSE value can be seen at the initial stage. High error values from all participating sensors at that region caused that sudden rise at the initial stage. Our proposed method can recover from that error situation faster and stabilize quicker than both IMC and Mean-based cleaning methods. The reason behind this is, IMC considers individual sensor reliability. We along with that look at the context relationship and mobility affinity between the candidate sensors for data cleaning. This facilitates better candidate selection for data cleaning and ensures more accurate data prediction. As a result, the difference between original data and predicted data is less resulting in achievement of lower RMSE value for our algorithm. Simulation

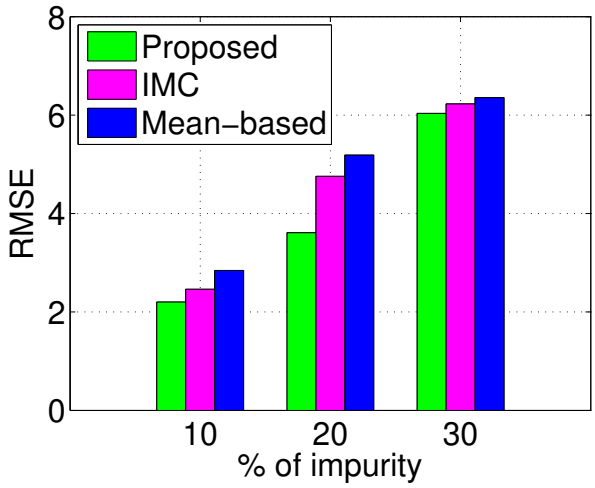



Figure 3.6: Performance comparison for different level of data impurity.

result shows up to 24% reduction in root mean square error (RMSE) over IMC and up to 30%compared to Mean-based cleaning during the considered time period

In Fig. 3.5 (b), we show RMSE comparison for nodes moving in variable speed and sensing humidity. Even though the node speed varies, our algorithm can predict the missing value with lower error than the compared algorithms. In Fig. 3.6, performance for a different level of impurity has been shown. The average RMSE incurred by our method was only 3.9. In the case of 10% impurity, our proposed method outperformed our IMC and Mean-based cleaning method by incurring on average 10.6% and 22.59% less RMSE, respectively. Our algorithm performs consistently by predicting the missing data value closest to the original value. For all tested level of data impurity, the level of error in missing data prediction was less for our algorithm in comparison to other algorithms.

## 3.5 Discussion and Summary

In this chapter, we proposed a novel mechanism for cleaning environmental sensing data streams that consider not only the sensed value, but also the sensor context and movement affinity for data cleaning. Because of the low quality and mobility around various environments, the data received from the tiny sensors are error-prone. We compared performance of our algorithm with a recently-proposed algorithm in cleaning data streams (i.e., IMC) and a widely-used method (i.e., Mean-based cleaning). We evaluated the approaches based on Smart City project in Melbourne factual dataset, to demonstrate our proposed method's efficacy in different rates of data impurity. The success of our approach lies in the integration of sensor context in correlated sensor identification for missing data prediction in environmental data streams. Our proposed method can predict the missing data value more accurately even when there is higher fluctuation in the data streams. The average RMSE incurred by our method was only 3.9 at varied data impurity rates. Simulation result shows up to 24% reduction in root mean square error (RMSE) over IMC and up to 30% compared to Mean-based cleaning during the considered time period.

CHAPTER 4

# SEMANTIC-AWARE TRAJECTORY DATA MINING

In chapter 3, we presented our context-aware data cleaning algorithm. Although the performance of our context-aware data cleaning method outperforms the related contemporary works (IMC [ZSS14] and mean-based [JAF+06, SGG10]) by achieving less root mean square error in the data prediction, there are nevertheless problems to address. The scalability test has not been performed. The algorithm was tested using simulated mobility traces. However, working with factual big data set is more challenging. The intrinsic 4V quality [CSK+14] of big data make data mining more intractable. Therefore, in this chapter, we develop a novel clustering-based method of trajectory data mining considering varied semantics of mobile trajectory. We organize this chapter as follows. In section 4.1, we discuss the background of this work. The detailed problem description is presented in section 4.2. Next, in section 4.3, we present different modules of our overall system and present algorithm for stop point identification and semantic-aware trajectory clustering. In section 4.4, we describe the experimental setup and discuss the evaluation results. Final discussion has been presented in section 4.5.

## 4.1 Introduction

Due to the growth of various mobile devices, there is also growth in mobile trajectories. As a result, there comes the research opportunity to find the movement relationship between these mobile nodes. In some relationships, only one entity is being considered; in others, inter-trajectory relationships are considered. For example, stop and go behavior can be found in a single trajectory. Some group movement behaviors include moving together, flock, and leader-follower. There ex-

ist some efforts on using the whole trajectory in order to find clusters of mobile nodes [NP06, VKG02].

We need to deal with an enormous amount of trajectory data generated by mobile devices that change their positions frequently to discover various movement relationships. In order to analyze their movement behavior, offline methods are not suitable. The consideration of the entire data stream might not be very useful, because the information to be extracted may be outdated at the time of processing [dSZdM16]. To handle this issue, we develop an online method that operates on the input data streams in real-time rather than traditional offline trajectory clustering and analysis methods. By looking at the recent time windows, we apply clustering on sub-trajectories to discover similarity in movement behaviors as well as identify outliers in real-time. Outliers have been defined as observations inconsistent with the remainder of that set of data [BL$^+$94]. The identification of outliers can lead to the discovery of useful and meaningful knowledge and has a number of practical applications (e.g., transportation, location-based services, public safety etc.).

Semantics is the piece of information that allows us to give a meaningful value to the data; for example, GPS coordinates consist of latitude and longitude, the semantic annotation will give those points a human-readable label such as "restaurant", "school", "bank" etc. Adding semantic annotation facilitates various movement and behavioral pattern in mobile trajectory data. In this chapter, we propose a Semantic-Aware Clustering-based (SACB) approach for trajectory data stream mining. We use DBSCAN [EKS$^+$96] as our baseline algorithm.

## 4.2 Problem Statement

The rapid development of mobile sensing technologies (e.g., GPS, accelerometer) embedded in smartphones has dramatically increased the number of positioning data (also known as trajectories). Trajectory data streams demonstrate several unique properties that together conform to the characteristics of big data (i.e., volume, velocity, variety, and veracity) and add challenges to data stream mining. A daunting challenge is to analyze movement behavior and detect outliers in rapid, voluminous streams of data. The research question that we want to answer in this chapter is: how can we identify the movement similarity and detect trajectory outlier in real-time?

Our hypothesis is that trajectory similarity based on time, space, direction, and semantics will facilitate real-time movement behavior analysis and trajectory outlier detection. We propose a method of adding geographic domain knowledge to raw trajectory data. Additionally, we propose a window-based online sub-trajectory clustering method for finding movement similarity based on space, time, direction and semantics.

## 4.3 Methodology

### 4.3.1 Add Geographic Domain Knowledge

In the raw data as shown in Table 4.1, only latitude and longitude values of the movement points are available. This module takes care of adding geographic domain knowledge to the points. As a result, the point receives a human understandable format. We add the type of place that the mobile object visited using this module. We consider the places that are in 15 m radius (which is common sensing range)

of the trajectory points. Table 4.2 presents the addition of geographic domain knowledge on the raw data as shown in Table 4.1. We used the reverse GeoCoding API [goo19] to get places within a pre-determined radius.

Table 4.1: Sample raw data

| Latitude | Longitude |
|---|---|
| 37.78574 | -122.4146 |
| 37.64401 | -122.45264 |
| 37.79434 | -122.39983 |

Table 4.2: Sample data after addition of geographic information

| Place Type | Latitude | Longitude |
|---|---|---|
| atm | 37.78574 | -122.4146 |
| restaurant | 37.64401 | -122.45264 |
| bank | 37.79434 | -122.39983 |

### 4.3.2 Distance Calculation

In this section, we measure the distance between sub-trajectories that are concurrently present in the considered time window $(t, t + |window|)$. We map the trajectory points (latitude, longitude) into Euclidean space in order to measure the distance between sub-trajectories in spatiotemporal dimension. If the sampling rates of a pair of sub-trajectories vary, we used linear interpolation to approximate the missing location point. The Euclidean distance between two objects that move within the same time window is calculated using Eqn. 4.1.

**Algorithm 2** CreateClusters

**Input :** Trajectories, range

```
1: ListOfClusters = []
2: for all point p ∈ Trajectory do
3:    for all clusterx ∈ ListOfClusters do
4:       if Latitude in range & Longitude in range then
5:          ListOfClusters[x].append(p)
6:       else
7:          ListOfClusters.newCluster(p)
8:       end if
9:    end for
10: end for
11: return ListOfClusters[]
```

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.1}$$

### 4.3.3 Direction Calculate

Forward azimuth is used to calculate the direction of movement for the trajectory points [Hed19]. Based on the calculated angle, the direction is defined. Different directions are : E, N, S, W, NE, NW, SE, SW, where 'E' is for the EAST direction, 'N' for the NORTH, 'S' for SOUTH, 'W' is for WEST, 'NE' is for NORTH-EAST, 'NW' is for NORTH-WEST, 'SE' is for SOUTH-EAST, and 'SW' is for SOUTH-WEST. The Cosine function is applied to the angle to calculate the directional distance.

### 4.3.4 Stop and Move Annotation

We find episodes in the trajectory. Based on the movement speed and duration of staying at a particular point, Algorithm 3 finds out the stop points in the trajectory. We have used a clustering-based method in order to annotate stops and moves in

**Algorithm 3** Find Stops

**Input :** Trajectory

```
1: ListOfClusters = CreateClusters(Trajectories, rng)
2: Stops = [ ]
3: for all point p ∈ Trajectory do
4:    for all cluster x ∈ ListOfClusters do
5:       if time in range & speed in range then
6:          stops[x].append(p)
7:       else
8:          moves[x].append(p)
9:       end if
10:   end for
11: end for
12: return stops[], moves[]
```

the trajectory. If the speed of a certain number of trajectory points is less than a threshold for a certain duration, then it is considered as a stop.

### 4.3.5 Group Formation

This module considers the semantic relationship between the mobile nodes. We follow a window-based mechanism in order to find out the group of trajectories that have similar behavior. There are a few steps that are maintained in order to identify the trajectories with similar behavior, such as:

(i) collect the trajectory data stream at each time window

(ii) apply a similarity measure

(iii) maintain the group(s)

(iv) discover the mobility

Algorithm 4 describes the formation of a group of trajectories that are similar based on time, space, direction and semantics. We use string matching in order to find the semantic distance. It is a window based density clustering algorithm where

we look at the sub trajectories of all moving trajectories that are present during that time period.

We use linear interpolation to align the points in time in order to measure the distance between a pair. The three weights for calculating distance has equal weight value in Eqn. 4.2 . If for some particular scenario spatiotemporal distance needs more emphasis than direction, then the weight value $w_1$ will receive a higher value. These three weight values are application dependent.

$$\begin{aligned} dist(SubT_{j,i}, SubT_{k,i}) = w_1 * distance(SubT_{j,i}, SubT_{k,i}) + \\ w_2 * direction(SubT_{j,i}, SubT_{k,i}) + \\ w_3 * semantic(SubT_{j,i}, SubT_{k,i}) \end{aligned} \quad (4.2)$$

Now, we briefly explain the Algorithm 4. First, we select a sub-trajectory of a moving object ($obj_j$, $SubT_{j,i}$) that is unvisited. It is a random selection. Then, we mark this sub-trajectory as visited and check if it can be a representative trajectory. The representative trajectory is identified following Gaussian kernel based voting method [PPK+12]. If it has minimum $\omega - 1$ neighbors in $\alpha$ radius, then a new group "g" is created including itself and the neighbors that are represented by ($obj_j$, $SubT_{j,i}$). Else, $obj_j$ is marked as ungrouped, because it cannot be represented by another representative that belongs to an existing group or it can be an outlier. At the end of the while loop, the ungrouped objects are declared as outliers. Algorithm 4 outputs a set of groups. By applying a post-processing mechanism on the groups, the leader among the group members is detected.

**Algorithm 4** Group Formation

**Input :** $S_i$ sub-trajectory set for the time $window_i = [t, t + |window_i|]$, $\alpha$ distance threshold, $\omega$ the size threshold, $\rho$ representativeness threshold

$G_i \leftarrow empty$
mark all the sub-trajectories in $S_i$ as unvisited
**while** $\exists(obj_j, SubT_{j,i}) \in S_i$ unvisited **do**
  mark$(obj_j, SubT_{j,i})$ as visited
  get neighborlist $N_\alpha(o_j)$
  **if** $|N_\alpha(obj_j) - 1| \geq \omega$ **then**
    **for all** $obj_k \in N_\alpha(obj_j)$ not visited **do**
      $dist \leftarrow$ Equation (4.2)
      **if** $e^{-dist^2 \div 2\sigma^2} > \rho$ **then**
        $g \leftarrow g \cup \{o_k\}$
        mark $(obj_k, SubT_{k,i})$ as visited
      **end if**
    **end for**
    $representative \leftarrow (obj_j, SubT_{j,i})$
    $G_i \leftarrow G_i \cup \{g\}$
  **else**
    mark $SubT_{j,i}$ as ungrouped
  **end if**
**end while**
**for all** $SubT_j, i$ ungrouped **do**
  mark $obj_j$ as outlier
**end for**
return $G_j$

## 4.4 Performance Evaluation

In this section, we describe our experimental setup and environment. We also describe the data set we used and different performance measures that we considered to evaluate the accuracy of our proposed method. We implemented our algorithm using the real dataset [PSDG09]. The dataset consists of GPS positions of taxi cabs that move around the city of San Francisco. The area of consideration has the size of 42.5 km X 62.5 km. There is a total of 160 taxis that were moving around different parts of the city. Fig. 4.1 depicts the raw point of movement taxis.

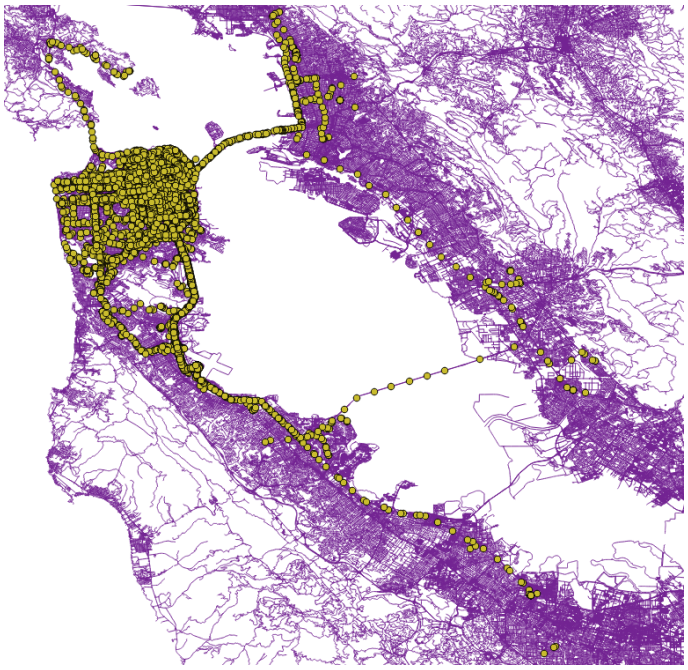


Figure 4.1: Taxi trajectory points on the map of San Francisco

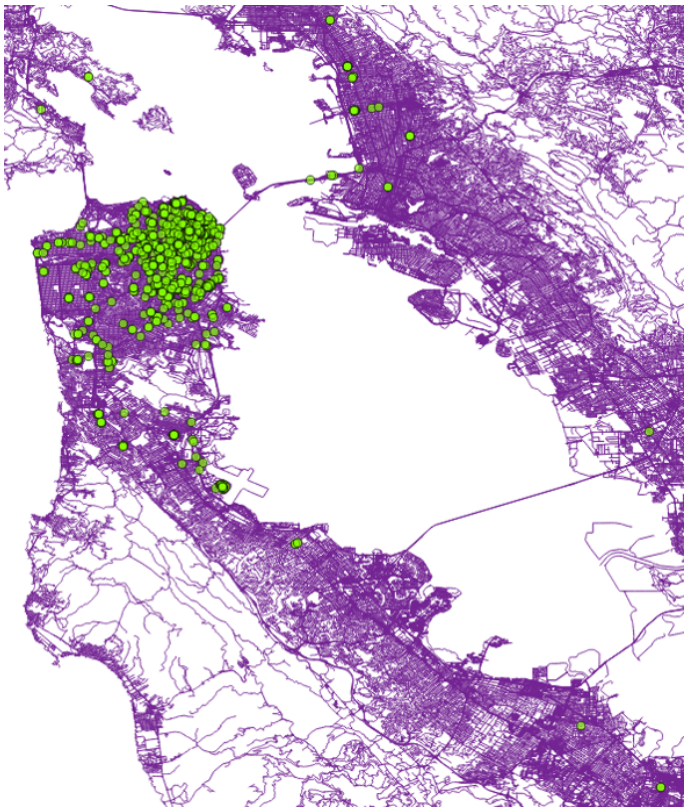


Figure 4.2: Stop points identified from all the taxi trajectories

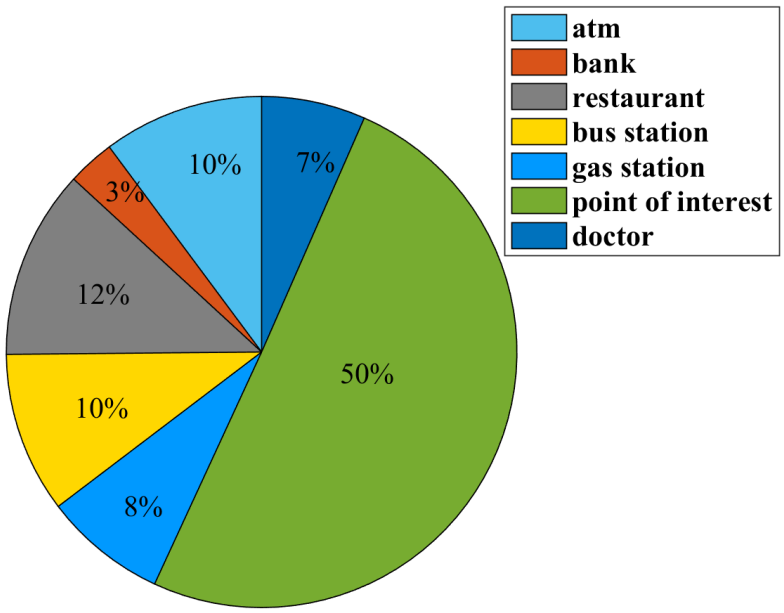



Figure 4.3: Stop points identified from all the taxi trajectories

After the clustering-based stop and go Algorithm 3, we get the stop and go episodes. We used the reverse GeoCoding API [goo19] to get places within a predetermined radius. After adding geographic information to the raw GPS points, we get the semantically annotated trajectory. These semantically annotated trajectories are used as input in Algorithm 3 and Algorithm 4. As a result, we can find stop episodes and group formation among trajectories, respectively. The stop points identified from all the taxi trajectories are shown on the map in Fig. 4.2.

Due to adding semantic information, we could classify the data based on weekend and weekdays. We analyzed the semantics of the stop points and the categorization on the weekend is shown in Fig. 4.3.

$$recall = \frac{SACB\_outlier \cap DBSCAN\_outlier}{|SACB\_outlier|} \tag{4.3}$$

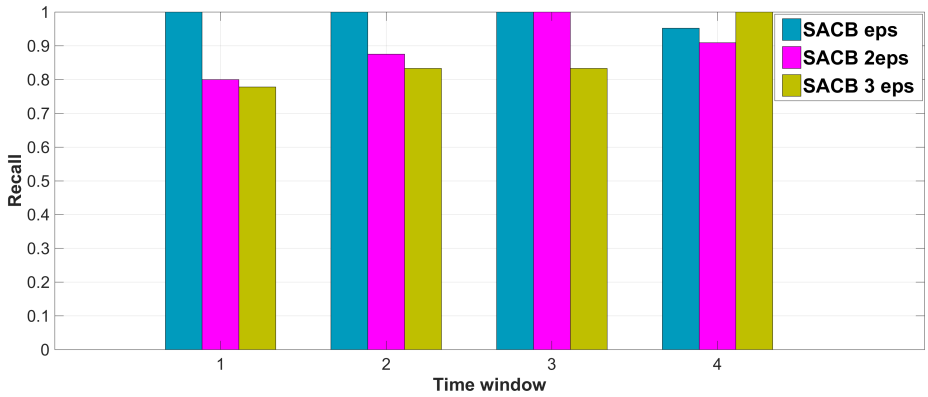



Figure 4.4: Recall for detecting outlier

$$precision = \frac{SACB\_outlier \cap DBSCAN\_outlier}{|DBSCAN\_outlier|} \tag{4.4}$$

After applying Algorithm 4, we identified the cluster of taxis as well as outlier taxis. The taxi that does not belong to any group is declared as an outlier. We captured timestamps from $9:42:00$ to $10:22:09$ in May $17, 2008$ as the experimental dataset to be used in implementing Algorithm 4. We set the time window as 5 minutes and tracked the clusters for 4 consecutive time windows. In Fig. 4.4, we show the performance of our algorithm in detecting outlier taxis. The recall values achieved during outlier identification at various time windows are shown. Three different clustering radius $(eps, 2eps, 3eps)$ have been tested. We used 250 as the eps value. Here, we consider the outliers identified by DBSCAN as the perfect list. The recall and precision is calculated using Eqn. 4.3 and Eqn. 4.4 [SL09]. $SACB\_outlier$ is the list of taxis identified as an outlier by our algorithm and $DBSCAN\_outlier$ contains the outlier taxis identified by the widely used density-based clustering algorithm: DBSCAN.

We tested with different density requirement $(\omega)$ and representativeness threshold $(\rho)$ values. In Fig. 4.5, we show recall values with respect to different node

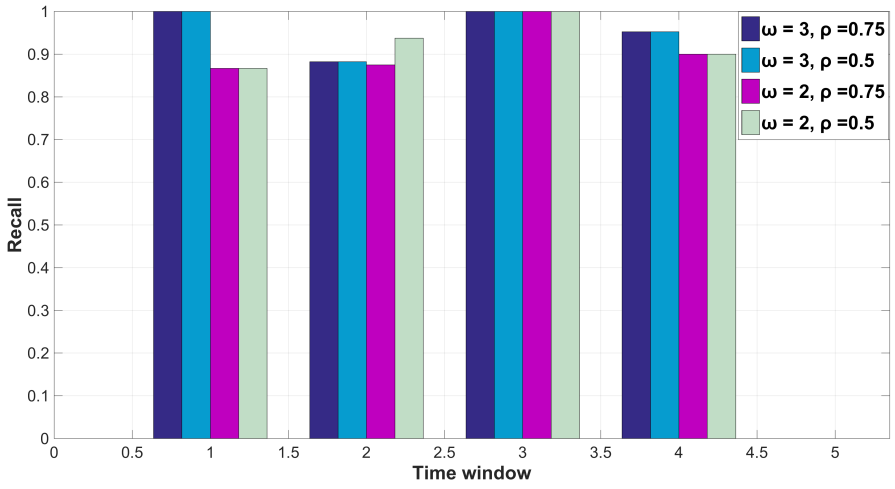



Figure 4.5: Recall for detecting outlier

density as well as representativeness threshold value. The range of $\rho$ is between 0 and 1.0. $\rho = 1$ means the highest similarity between a pair of objects, the best candidate to be declared as representative among the group members and later leader in the moving group. Even in high $\rho$ value, which demands close correspondence among the trajectories based on their spatiotemporal, directional and semantic relationship, our method was able to detect outlier taxis almost similar manner as DBSCAN by maintaining recall value over 0.85. The average recall value achieved was 0.92 for the different combination of $\omega$, $\alpha$ and $\rho$.

Fig. 4.6 shows the F1 score achieved at time window $1-4$. F1 score is calculated using eqn. 4.5 [SL09]. The average F1 score encountered thoughout the experiments was 0.8, which indicates the classification accuracy of our $SACB$ method.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.5}$$

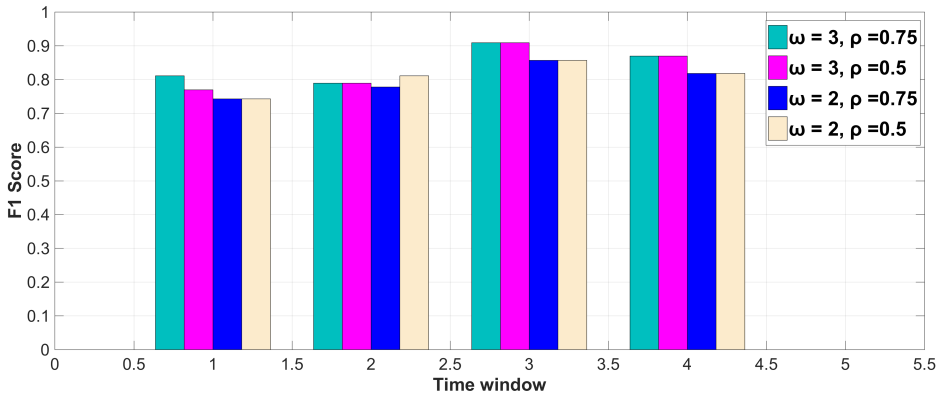



Figure 4.6: F1 score for detecting outlier

## 4.5 Discussion and Summary

In this chapter, we introduced a semantic-aware online clustering-based method for finding movement relationships in mobile trajectories. We used semantic annotation to annotate the movement trajectories. After adding the geographic information to raw GPS points, window-based online clustering is applied to find sub-trajectory groups. Ours is an online method that performs the operation on the input streams in real-time rather than traditional offline trajectory clustering and analysis methods. We conducted experiments on a real-world data set. Along with the added advantage of semantic-aware movement behavior analysis, our method was able to successfully identify outliers in the clustering process with an average recall of 0.92 and F1 score of 0.8.

CHAPTER 5

# CORRELATED SENSOR-BASED DATA FUSION IN MOBILE CROWDSENSING

In the previous chapters, we assumed that all mobile devices are honest. They co-operate in the data collection procedure. However, malicious participants can report false data in mobile crowdsensing applications. Also, the sensors used in the previous chapters were homogeneous, they sensed similar data. In contrast, nowadays due to technology advancement, a smartphone contains varied sensors. To exploit these resources intelligently in the missing data prediction, in this chapter, we develop a data fusion mechanism utilizing the inter sensor-category correlation. We organize this chapter as follows. In section 5.1, we discuss the background information and challenges. In section 5.2, we discuss the problem and our hypothesis. In section 5.3, we describe different modules of our overall system. Next, in section 5.4, first, we discuss the simulation setup, and then we present our results and analysis. Finally, discussion and summary are offered in section 5.5.

## 5.1 Introduction

With the advent of better wireless technology and an increase in smartphone usage, a new mode of data collection named mobile crowdsensing (MCS) has emerged. Mobile crowdsensing has a number of practical applications: traffic monitoring, epidemic disease monitoring, reporting from disaster situations and environment monitoring [CDW+15], [FZ16], [RGB+17]. For example, an environmental air quality sensing system was deployed on street sweeping vehicles to monitor air quality in San Francisco [AHM+09]. These applications are usually open to the public and receive sensor data from multiple participants. This influences the reduction of data sparsity at lower costs in comparison with traditional sensor networks. With

various advantages, MCS's people-centric architecture allows both more inaccurate and corrupted data [MMH+15]. Malicious participants can manipulate the MCS data collection process at ease. These entities can interrupt a system by reporting fabricated or erroneous data, making trust evaluation a highly important issue in MCS applications. Therefore, validating the accuracy of contributions is essential to ensure the reliability of the application system.

In this paper, we consider data corruption attack behavior of a malicious participant. By malicious we mean a participant who sends incorrect data either intentionally or unintentionally. The unintentional error can arise because a participant carelessly performed the sensing task, or due to a sensor error. On the contrary, a malicious participant can deliberately fabricate the sensed data to infiltrate the system. For example, in air quality monitoring, a malicious participant may hold the sensor beside a burning cigarette or place it over sand instead of facing to the air. Thus, the reported data will not represent the actual air quality. In the related contemporary works [KLM16], [Kou18], [SPP+06], [TPI17], the authors did not consider the participants' malicious behavior. Thus, these works were not able to distinguish the sensing data reported by malicious or careless users. This limitation of the existing works motivates us to design reputation-aware real-time data fusion algorithms for MCS to ensure data integrity. Our method can detect malicious participants and prevent them from infiltrating the system in real time.

We develop an online method for data quality prediction in MCS considering the heterogeneous trust level of the participants. We took into account spatiotemporal and inter sensor-category correlations. We consider the users who are willing to participate in sensing at the same time. The terms *participant* or *node* are used to denote a user with sensing capability.

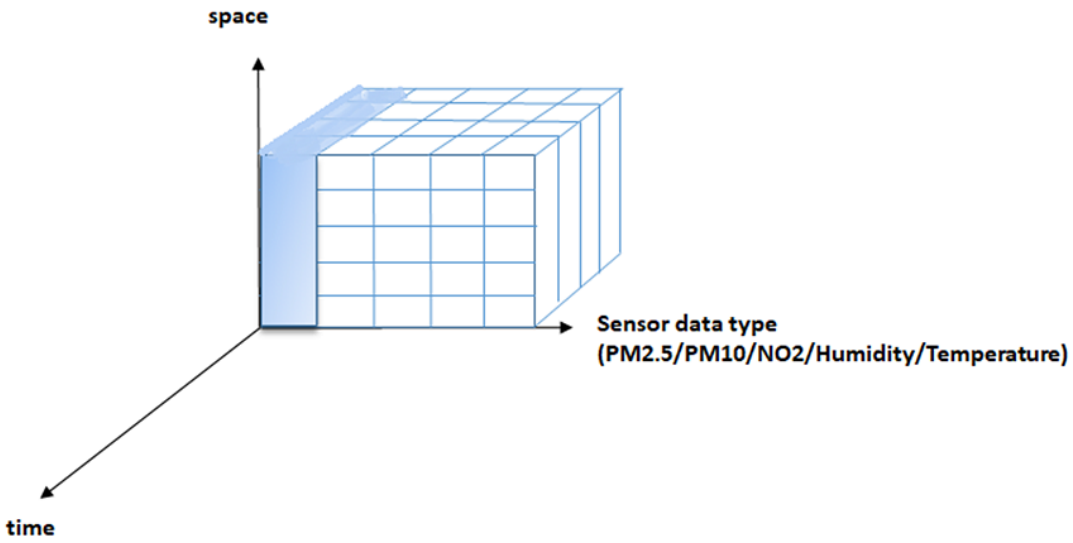



Figure 5.1: Three-dimensional Tensor

We implement our Correlated Data and Reputation-aware Data Prediction (CDR) method on two real-world datasets [BBL+14], [ZLH13]. The sensing was performed for four days, and there are 289 taxi values in the first real dataset. The taxis move around different parts of Rome sensing temperature. The second data set consists of Beijing's air quality data. One hundred and forty-nine taxis with four types of sensors collect $PM2.5, PM1.0, NO_2$ and *humidity* data from Beijing for seven days.

## 5.2 Problem Statement

Smartphones are an example of a contemporary heterogeneous sensor network. They are equipped with different sensors like humidity, light, accelerometer, and proximity sensor. In the presence of varied sensors, designing an appropriate fusion mechanism is essential. The goal of this chapter is to develop a method of accurately estimat-

ing a missing value from incomplete sensor data. Therefore, we plan to answer the following question: How to utilize the correlation between different sensor types to improve the accuracy of the data fusion mechanism?

We investigated different factual data sets to observe the correlation between different data types. For example, it can be observed from an air quality monitoring application that it contains particulate matter with a diameter under $2.5\mu m(PM2.5)$, $PM1.0$, $NO_2$, temperature, pressure and humidity data. Thus, observing the correlation between these sensor types and utilizing in the development of data fusion mechanism is the goal of this chapter. Our hypothesis is exploiting sensor category correlations in the Least square fit-based data fusion mechanism will facilitate better data prediction by reducing the data prediction error.

## 5.3 Methodology

In this section we first present an overview of the proposed mechanism, Correlated Data and Reputation-aware Data Prediction (CDR), then a detailed description of the components, and finally how we fit them together to create our full structure.

### 5.3.1 Overview

CDR consists of two parts: a reputation calculation method and correlated data [KLM16]. The reputation method considers two types of trust for each sensor, *cooperation* and *reputation*, and both parameters are calculated at the application server level. The reputation calculation method is applied to multiple types of sensor data streams. These varied sensors are correlated with each other. It is important for our mechanism to take the granularity of time and space into account. We discretized our time into epochs, and space into equal-sized grids. The framework is

applied only on data from sensors within the same *region* and the same epoch. CDR is applied to each different *type* of data and then the final, discretized space-time blocks are used to produce a least-square regression on the target data type. This regression can be used to predict both future data and missing data. We borrow the concept of three-dimensional tensors shown in Fig. 5.1 from [KLM16]. The authors considered temporal interpolation for the sparse regions. However, Kang et al. [KLM16] assumed that all incoming data from sensors was accurate.

### 5.3.2 Cooperation

Cooperation scores of sensors are measured per epoch; they measure the proportion of the inverse square root error of the data from the sensor over the sum of the proportion of the inverse square root error from all sensors. For our cooperation parameter, we used an inverse proportion of the square root of the absolute error so as not to punish small deviations from the average as much. In the data sets we tested, temperature data and air quality data, small variations from the average are common. The equation for cooperation score is shown in Eq. 5.1.

$$p_i = \frac{\frac{1}{\frac{\sqrt{|x_i - r|}}{\sum_{i=1}^{n}\sqrt{|x_i - r|}}}}{\sum_{j=1}^{n}\frac{1}{\frac{\sqrt{|x_j - r|}}{\sum_{i=1}^{n}\sqrt{|x_i - r|}}}} \tag{5.1}$$

Where $r$ is the *robust average* of the data in that epoch and $x_i$ is the measurement from sensor $i$. The *robust average* of the data provides an idea of where the data clusters, and this increases the accuracy of the data by assigning more weight to values that occur more frequently. We calculate *robust average* using Eq. 5.2.

$$r = \sum_{i=1}^{n} p_i * x_i \tag{5.2}$$

### 5.3.3 Reputation

Reputation scores are updated at the end of each epoch; it measures how accurate the crowdsensing participant has been over time. To calculate reputation from cooperation scores, first, the cooperation scores are normalized [HKH14] using Eq. 5.3. Here, $P_i$ is the cooperation score of participant $i$. $min(p)$ and $max(p)$ denote the minimum and maximum cooperation score among all the participants during that epoch. After normalization, the cooperation scores belong to the range $[-1, 1]$.

$$p_i^{norm} = \frac{2(p_i - min(p))}{max(p) - min(p)} - 1 \tag{5.3}$$

We want to maximize the impact of the most recent epochs and minimize the impact of the least recent ones. To make the aging effective, we *age* the normalized cooperation scores with Eq. 5.4.

$$p_{i,k}^{'} = \sum_{k'}^{k} \lambda^{k-k'} p_{i,k}^{norm} \tag{5.4}$$

Here, $k$ denotes the current epoch and $k'$ has the value from 1 to current epoch. Aging parameter $\lambda$ has the value $[0, 1]$ Finally, reputation is calculated using the Gompertz function [HKH14], shown in Eq. 5.5.

$$R_{i,k} = ae^{be^{cp_{i,k}^{'}}} \tag{5.5}$$

### 5.3.4 Full Structure

We discretize the space into *regions* and the time into epochs, then we run CDR on every discrete block of space-time.

First, we run an Expectation Maximization Algorithm (EM), shown in Algorithm 5, on the "reputable" sensors. To be classified as reputable sensors, the participant

must have a reputation higher than the threshold. This threshold is an application dependent. Initially, all sensors are classified as reputable with equal cooperation score.

**Algorithm 5** Expectation Maximization on Cooperation Scores for Robust Average

**Input:** Robust Average ($r$) , Cooperation Scores ($p_i$)
**Output:** Robust Average ($r$)
*Initialize:* all $p_i$ to $1/n$, where $n$ is the number of sensors, and $l = 0$, where $l$ is the iteration
**while** $p_i^l$ *and* $p_i^{l+1}$ *don't converge* **do**
  Compute $r^{l+1}$ from $p_i^l$'s using Eq. 5.2
  Compute $p_i^{l+1}$'s from $r^l$ using Eq. 5.1
  $l = l + 1$
**end**
**return** $r^{l+1}$

After running the EM algorithm once on only the reputable sensors, we then check the reported values from "disreputable" sensors, or sensors with a reputation lower than the threshold. If the reported value from any of these sensors is within an acceptable error range of the *robust average* calculated from the reputable sensors' reported data, then it is added as a faux reputable sensor in that block of space-time. After finding all the sensors from the set of disreputable sensors that contributed acceptable data in the block of space-time, EM is then run again on the new set of reputable sensors. The reason that we run EM twice is to provide sensors in the disreputable set a chance to move into the reputable set if they consistently contribute accurate data because only sensors with a cooperation score for the epoch will have their reputations updated. The second EM run gives a new reputable average as well as update reputation scores for each sensor.

The new reputation scores are then normalized to the range $[-1, 1]$ using Eq. 5.3. The normalized cooperation scores are then aged based on their cooperation rating. Sensors with a cooperation score above a certain threshold are labeled as

"cooperative" and sensors with a cooperation score below that threshold are labeled as "uncooperative". Depending on the sensor's classification for the latest block, the normalized cooperation is multiplied by a different aging parameter, $\lambda$. Cooperative sensors are multiplied by a lower aging parameter than uncooperative sensors. This means that the growth and decay rates of reputation will be different; the decay rate will be higher, and this provides higher punishment for bad data and thus helps quickly detect malicious users. Finally, the aged cooperation score is inputted to Eq.5.5.

Once all the blocks are processed for each data *type*, then we use the processed data to create a least-square fit with the non-target data as the coefficient matrix, $A$, and the target data type as the dependent matrix, $b$ as shown in Eq. 5.6.

$$\mathbf{A}\hat{x} = \mathbf{b} \tag{5.6}$$

The regression, $\hat{x}$, is then used to predict the target value given knowledge of all the other data values.

## 5.4 Performance Evaluation

We used the percentage absolute difference and Root Mean Square Error ($RMSE$) as performance metrics of data prediction accuracy. We compared the performance of our CDR method against mean-based and temporal linear regression-based data prediction models. We tested using two real-world data sets. In the first data set, our target type is temperature and uses two types of simulated correlated data. In the second data set, our target type is particulate matter with a diameter under 2.5 $\mu m$ ($PM2.5$) and uses three types of real correlated data ($PM1.0, NO_2$ and *humidity*).

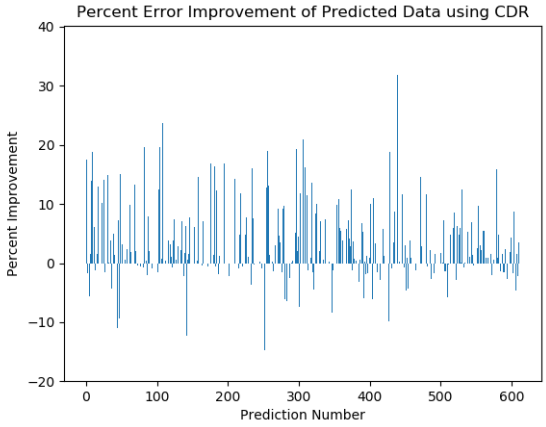



Figure 5.2: Prediction results for test set 1: out of 612 predictions, CDR performed better in 466 and was within 5% of the true value in 290 cases

## Temperature

The temperature data was from an area of roughly $22km$ by $23km$ and was taken over four days. The experimental area was split into 25 *regions* using a 5x5 equal-sized grid. We split the execution time into 96 epochs with each epoch being one hour long. We tested the performance of our CDR method against the existing mean-based method in three test data sets. To imitate the data impurity, continuous or random errors were applied on the temperature data streams. The data error from malicious participants ranged from 25% to 75%. Figures 5.2 through 5.4 show CDR's percentage improvement over the mean-based method, and each figure shows 612 predictions.

On average CDR was 16% more accurate and performed better in 77 percent of cases. Our CDR method incurred a cumulative percentage error of 9.3%.

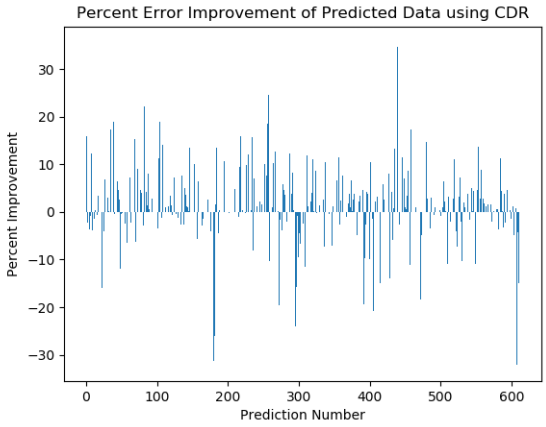



Figure 5.3: Prediction results for test set 2: out of 612 predictions, CDR performed better in 453 and was within 5% of the true value in 261 cases

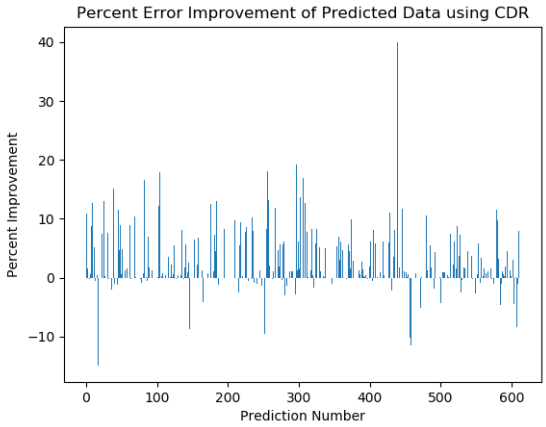



Figure 5.4: Prediction results for test set 3: out of 612 predictions, CDR performed better in 498 and was within 5% of the true value in 213 cases

## PM2.5

The air quality data was collected from an area of roughly $120km$ by $150km$. The duration was seven days (149 hours). CDR was tested against the existing mean-based and temporal linear regression-based data prediction methods on five test data sets. To imitate the data impurity, continuous or random errors were applied on the crowdsensing data streams. The data error ranged from 25% to 75%.

We tested the performance of our algorithm for different levels of erroneous data from malicious users. We also varied the knowledge level of the participants in regards to the experimental environment to imitate sophisticated data manipulation by a malicious crowdsensing participant. The test set 1 (Fig. 5.5, Fig. 5.10, Fig. 5.15) was used for missing data prediction. We tested with sequential and random data loss patterns. In the first experiment with erroneous data from malicious users (Fig. 5.6, Fig. 5.8, Fig. 5.11, Fig. 5.13, Fig. 5.16, Fig. 5.18), we assumed the participants did not have any prior knowledge about the experimental environment. The data error ranged from 25% to 75%. One group of malicious participants reported a fixed percentage of error throughout the experiment. In the second experiment, we considered that the malicious participant has extended knowledge about the sensing area (Fig. 5.7, Fig. 5.9, Fig. 5.12, Fig. 5.14, Fig. 5.17, Fig. 5.19). Thus, these participants try to change the sensing data by adding noise to the air quality data of that particular spatiotemporal unit.

### Percent Error per Prediction

Figures 5.5 through 5.9 show CDR's percentage improvement over the mean-based method, and each figure shows 640 predictions. On average CDR performed better in 70% of cases and is 70% more accurate.

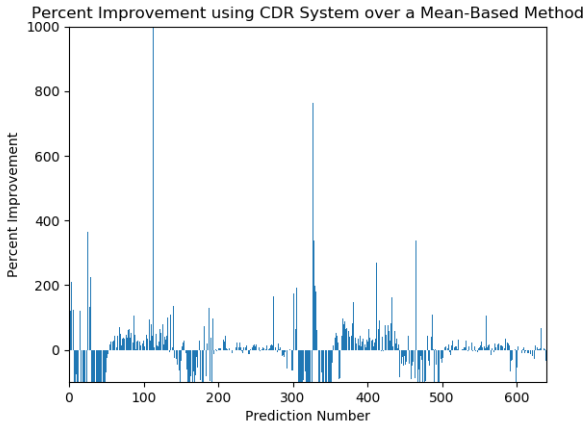



Figure 5.5: Prediction results for test set 1: out of 640 predictions, CDR performed better in 379 cases

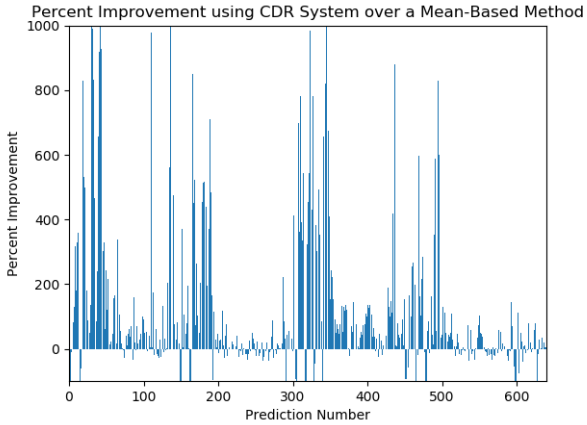



Figure 5.6: Prediction results for test set 2: out of 640 predictions, CDR performed better in 445 cases

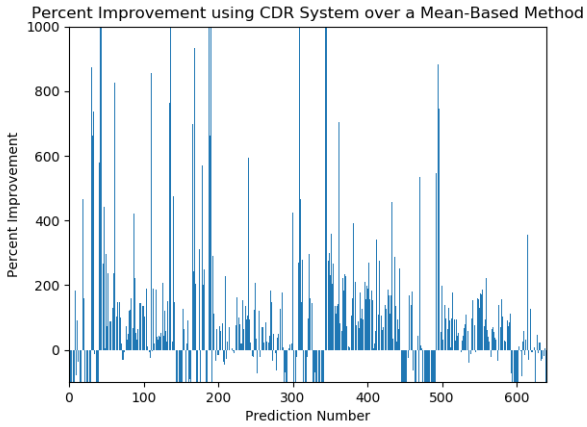



Figure 5.7: Prediction results for test set 3: out of 640 predictions, CDR performed better in 442 cases

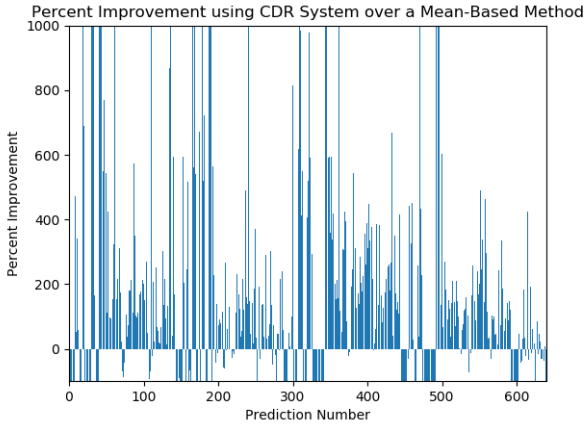



Figure 5.8: Prediction results for test set 4: out of 640 predictions, CDR performed better in 454 cases

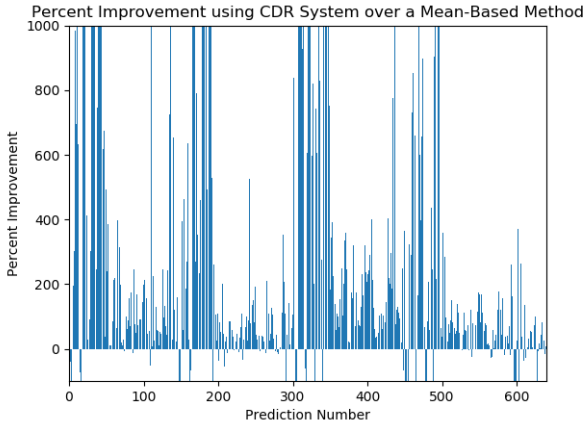



Figure 5.9: Prediction results for test set 5: out of 640 predictions, CDR performed better in 533 cases

### Root Mean Square Error by Epoch

Figures 5.10 through 5.19 show CDR's improvement of the root mean square error ($RMSE$) normalized by epoch. We calculated $RMSE$ and used it as a performance measurement criteria of our algorithm. $RMSE$ is a standard metric to evaluate the accuracy of the prediction model [ZSS14].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{V}_i - V_i)^2}, \tag{5.7}$$

where $\widehat{V}_i$ is the predicted value, $V_i$ is the original value and $n$ is the number of epochs.

On average CDR had a lower $RMSE$ than the mean-based method in 64 percent of the epochs and had a lower $RMSE$ by 25%. CDR's average $RMSE$ was 0.66, the average value of the target data type, $PM2.5$, was 79 with a range of $[4, 244]$.

Figures 5.15 through 5.19 show CDR's improvement in $RMSE$ over a temporal linear regression-based data prediction model. On average CDR incurred a lower

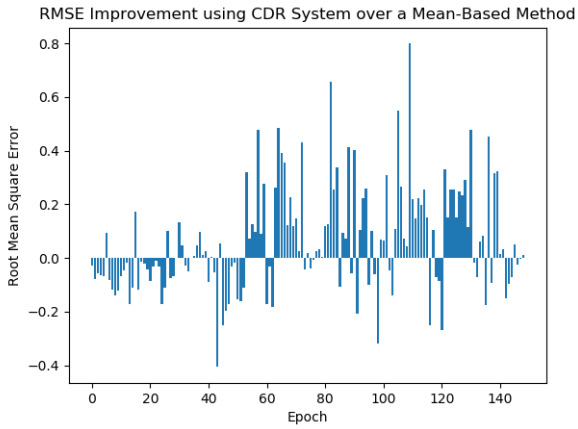



Figure 5.10: Prediction results for test set 1: out of 149 epochs, CDR performed better in 88 epochs

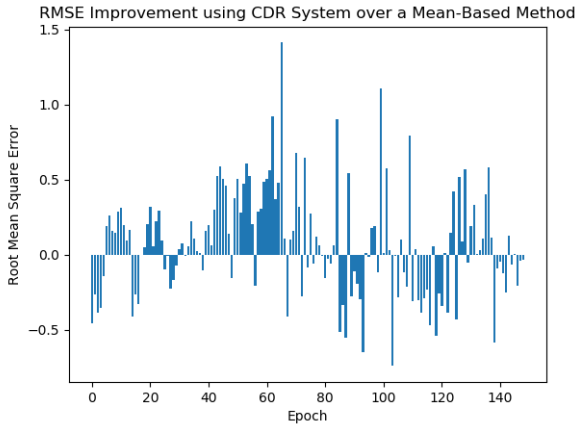



Figure 5.11: Prediction results for test set 2: out of 149 epochs, CDR performed better in 88 epochs

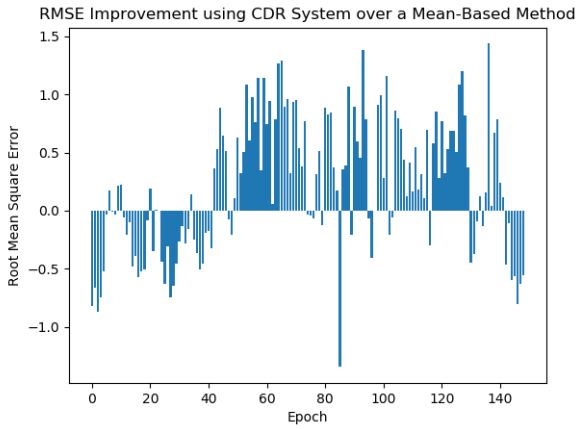



Figure 5.12: Prediction results for test set 3: Out of 149 epochs, CDR performed better in 90 epochs

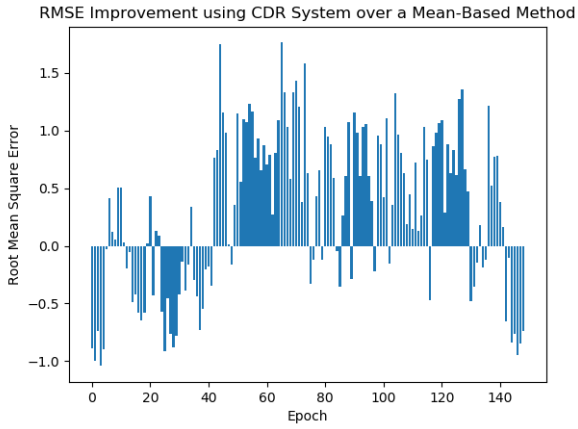



Figure 5.13: Prediction results for test set 4: Out of 149 epochs, CDR performed better in 96 epochs

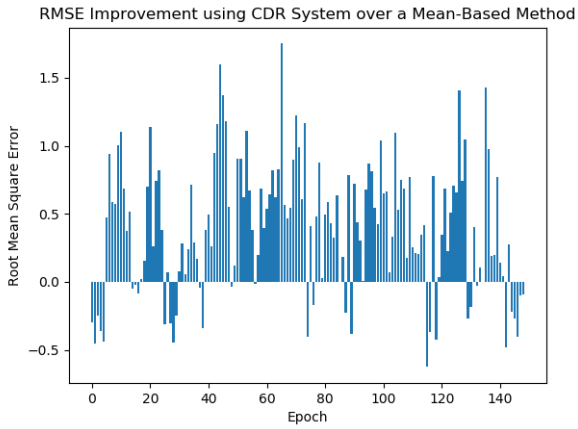



Figure 5.14: Prediction results for test set 5: out of 149 epochs, CDR performed better in 115 epochs

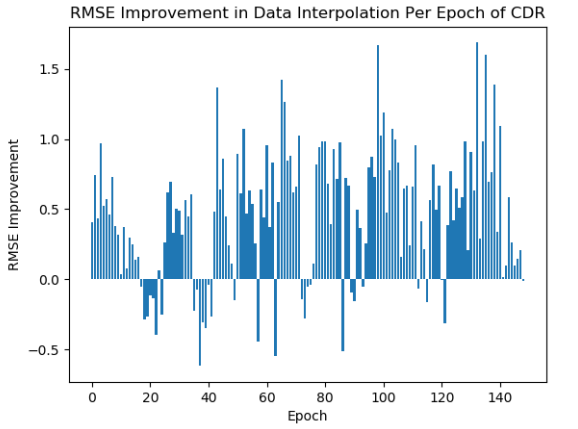



Figure 5.15: Prediction results for test set 1: out of 149 epochs, CDR performed better in 119 epochs

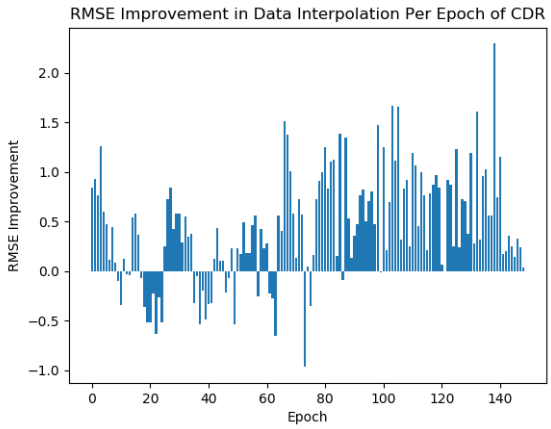



Figure 5.16: Prediction results for test set 2: out of 149 epochs, CDR performed better in 119 epochs

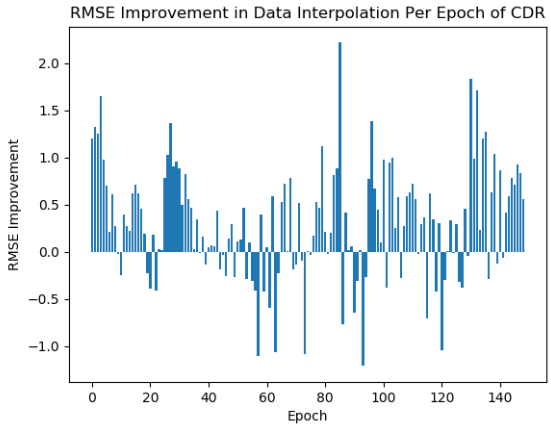



Figure 5.17: Prediction results for test set 3: out of 149 epochs, CDR performed better in 105 epochs

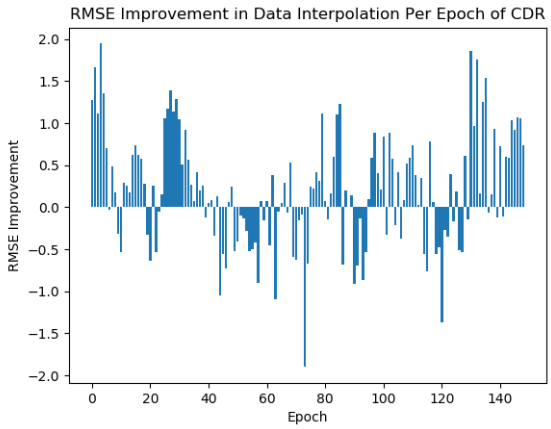



Figure 5.18: Prediction results for test set 4: out of 149 epochs, CDR performed better in 93 epochs

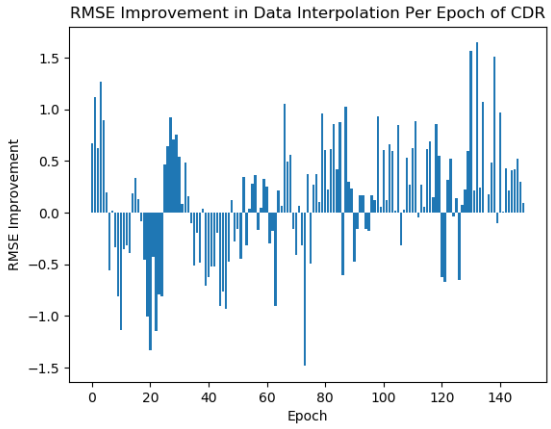



Figure 5.19: Prediction results for test set 5: Out of 149 epochs, CDR performed better in 95 epochs

$RMSE$ than the linear regression model by 59%, and performed better in 71 percent of epochs.

## 5.5 Discussion and Summary

In this chapter, we proposed a novel method, named as CDR, for reputation-aware data fusion for mobile crowdsensing data streams. We showed that the proposed mechanism outperforms the existing mean-based and temporal linear regression-based data prediction models. We evaluate the approaches based on two datasets: Rome crowdsensing temperature and Beijing Air quality datasets, to demonstrate CDR's efficacy in different scenarios. For the Rome crowdsensing dataset, we achieved 16% better accuracy. Specifically, the 9.3% prediction error in temperature measurements of our approach equates to roughly 1 degree difference, which is negligible in real-life applications. With this in mind, we can say that our mechanism predicts temperature values with high accuracy. In case of the air quality dataset, our CDR method incurred on average 25% and 59% less $RMSE$ than mean-based and temporal linear regression models, respectively. Our data fusion method incurred an average $RMSE$ of 0.66 per epoch, which insinuates higher data prediction accuracy. The success of our approach lies in the integration of dynamic trust evaluation of the sensed data which allows us to defend data corruption attack and identify malicious or honest participants based on their reported data in real time.

CHAPTER 6

# REPUTATION-BASED CONTEXT-AWARE DATA FUSION RESILIENT AGAINST MALICIOUS ATTACKERS IN IOT

In chapter 5, we presented a data fusion method utilizing inter sensor-category correlations. However, the on-off attack behavior of a malicious participant was not considered. Also, it was assumed that the majority of the participants are honest. In the experiments, the method could tolerate up to 30% of malicious participants. Moreover, equal-sized grid-based spatial discretization was used in the previous chapter. In this chapter, we consider dynamic spatial granularity which is suitable for location sensitive applications (caused due to diffusion and dispersion). Moreover, the participants did not have additional knowledge. To overcome the limitations, in this chapter, we develop a context-aware reputation-based data fusion method that is resilient against on-off and data corruption attackers. Even in the presence of a high number of sophisticated malicious participants, the proposed method is able to ensure data reliability in the mobile IoT application. We organize this chapter as follows. In section 6.1, we discuss the background information and challenges. The problem formulation is presented in section 6.2. In section 6.3, we discuss different modules of our overall system and present the reputation and trust distribution mechanisms. After that, the reputation-aware missing data prediction methodology is presented in section 6.4. In section 6.5, we discuss our performance evaluation. First, the simulation settings and then the results and analysis are discussed in section 6.5. Finally, discussion and summary are offered in section 6.6.

## 6.1 Introduction

Mobile internet of things (IOT) has been renowned as a state-of-the-art sensing data gathering epitome [LZW+17]. The rapid development of mobile sensing technologies

(e.g., smart devices embedded with various powerful sensors such as temperature, accelerometer, humidity, and gyroscope) has encouraged the proliferation of Internet of things (IoT). Mainstream smartphones and recently popular wearable devices such as smart watches, fitness tracking devices, and narrative clip are equipped with many sensors. These devices can be used as abundant sources of raw data [HXL+17]. In recent years, mobile IoT has gained increased applications in different areas, including transportation, air quality monitoring, epidemic disease monitoring, reporting from disaster situations, environmental monitoring and so on [CDW+15], [FZ16], [RGB+17], [PK19].

HazeWatch [SCHL13], a mobile IoT system, depends on citizen participation for air pollution monitoring. Air pollution has a negative impact on public health. As per the statistics published by World health organization (WHO), 4.2 million premature death occurs annually due to air pollution. High concentrations of particulate matter with a diameter less than 2.5 $\mu$ m (PM2.5) in the pollutant air causes cardiovascular or respiratory diseases, and cancers. Unfortunately, most people across the world, specifically 91%, inhabit areas where the air pollution levels exceed WHO defined limits [Org18]. Agencies such as National Environment Agency of Singapore are now using HazeWatch every day.

Different from the traditional sensor networks, where a large number of sensors are required to be deployed to sense data, mobile crowdsensing is open in nature, allowing anyone to participate at any time. In mobile crowdsensing-based IoT applications, the task of sensing is assigned to a person. However, successful information transmission largely depends on multiple factors. Some of these factors are behavioral (lack of time or willingness), and others are due to the resource limitation (e.g., network bandwidth and smartphone battery) for performing the sensing task. Also, this people-centric architecture allows both more inaccurate and corrupted data

[MMH+15]. Malicious participants can easily manipulate the IoT data collection process by reporting fabricated or erroneous data.

Although reliability and accuracy are of utmost importance in many sensor applications (e.g., air quality monitoring), it is often difficult to ensure these properties in such applications. In the air quality monitoring application, the sensed pollution data is used to create a pollution map. Imprecise pollution information about an area will mislead people. For instance, due to erroneous pollution data, an asthma patient who prefers a pollution-free route for a walk might be directed to a polluted area. This misdirection will have a negative impact on his/her physical well-being. Thus, trust evaluation is a major issue in these applications to ensure data reliability and integrity. Data reliability refers to the condition when data attains enough completeness to be considered for its goal and context [MW04].

While researchers [GLN15],[LZW+17],[GL15],[ACFM14],[KWW+16], [HKH14], [TPI+18] have attempted to improve the quality of the received sensor data, limited research has been done on how sensor context (e.g., sensor model, terrain elevation, wind speed, population density, and user movement during sensing) can be used in sensor selection for data cleaning. In related contemporary work [GLN15],[LZW+17], [TPI17], [GL15], the authors considered user context for data quality estimation in mobile IoT. However, Gill et al. considered either temporal [GL15] or spatial relationship [GLN15] among the sensors while developing model-based data cleaning mechanisms. Because only one type of relationship is considered, these methods did not achieve decent cleaning accuracy and have limited practical impact. The methods failed to exploit the dynamism of the experimental environment while selecting the correlated sensors for data prediction. On the other hand, the authors [LZW+17] considered the presence of exactly one mobile user at each point of interest (PoI) which is a limited setting and not practical in real-world scenarios. Further-

more, the authors did not consider the participants' malicious behavior. Thus, these works were not able to distinguish the sensing data reported by malicious or careless users. This limitation of the existing works motivates us to design context-aware reputation-based real-time data fusion algorithms for MCS to ensure data integrity. Our method can detect malicious participants and prevent them from infiltrating the system in real time. Even in the case of high false data injection, our method is able to ensure data reliability.

In this chapter, we consider on-off and data corruption attack behavior of a malicious participant. Data corruption attack occurs when a participant sends incorrect data either deliberately or recklessly. The reckless error occurs when a participant heedlessly performed the task of sensing or was caused by a sensor error. In contrast, a malicious participant can intentionally fabricate the sensed data to penetrate the system [TPI+18]. For example, in air quality monitoring, a malicious participant may hold the sensor beside a burning cigarette or place it over sand instead of facing to the air. Thus, the reported data will not represent the actual air quality. In contrast, on-off attack means that malicious participants behave good and bad alternatively, hoping that they can remain undetected while hampering data quality [SHYL06], [LGS15].

In this chapter, we develop an online method for data quality prediction in mobile IoT considering the spatiotemporal, contextual and inter sensor-category correlations among the participants. We consider the users who are willing to participate in sensing at the same time. The inter-node distance at a specific instance, as well as user context (sensor model, wind speed, and user movement during sensing), are considered in correlated node selection. The terms participant, user or node are used interchangeably to denote a user with sensing capability. We considered different spatial granularity while defining the correlated participants for data fusion depend-

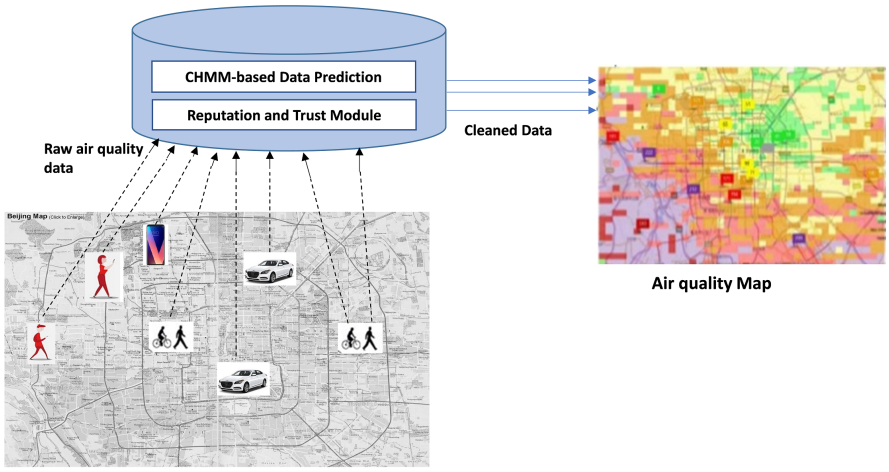



Figure 6.1: Overall Architecture: Air quality Monitoring application

ing on the application types: spatial stable or variable, which was not considered in the earlier works. Our methodology is resilient against data corruption and on-off attack behavior of a malicious participant. We implement our Reputation-based Context-aware Data Fusion method on Beijing's air quality dataset [ZLH13]. One hundred and forty-nine taxis with four types of sensors collect $PM25, PM10, NO_2$ and *humidity* data from Beijing for seven days. We are the first to use Contextual Hidden Markov Model (CHMM) for online data prediction in mobile crowdsensing for the IoT. CHMM has the unique capability of fusing temporal dependence and contextual spatial relationships [ZWL07], [ZAS16], [CB97]. Also, it is a light-weight method, and thus suitable for our big data application. This motivates us to apply CHMM for data prediction in mobile IoT. We compared the performance of our RCoD method against four state-of-the-art works and the results justify its superiority.

## 6.2 Problem Statement

There is data imprecision or missing values in the crowdsensing applications due to frequent loss of communication, hardware error or malicious intention of the carrier. Thus, it is important to detect those data imprecision and predict those incorrect and/or missing values. The problem being addressed in this chapter is predicting data in a more accurate manner in the presence of malicious participants who inject false data to vandalize the system. Our goal is to recover those missing or imprecise data values from the correlated data streams.

Let us assume that there are $N$ participants identified as trusted from the reputation system. Thus the whole data matrix has the size $N \times T$, where $T$ is the duration. Matrix $V^{(N \times T)}$ represents all time series ($T$) values from the $N$ trusted participating sensors. Matrix $E$ keeps track of the missing data. If there is a missing value or erroneous reading ($V_{i,j}$) from $i^{th}$ participant during a particular timestamp $j$ , $E_{i,j} = 1$, otherwise $E_{i,j} = 0$. The size of the error matrix is $N \times T$, the same as $V$. The problem of missing value prediction is defined as follows.

Given $M = \{V, E, C\}$, estimate $\tilde{V}_{ij}, for(i,j) \in (i,j) : E_{i,j} = 1$. Where $V \in \mathbf{R}^{N \times T}$ represents the $T$ timeseries data from $N$ crowdsensing participants, $E \in \mathbf{R}^{N \times T}$ represents the error matrix and $C \in \mathbf{R}^{N \times N}$ is the contextual matrix. The matrix $C$ denotes the pairwise contextual correlation among the participants. The data range of matrix $C$ is $[-1, 1]$. Here, 0 denotes no correlation and a higher value insinuates higher correlation. Hence, it has a size of $N \times N$. Our hypothesis is a context-aware reputation-based data fusion mechanism will facilitate the accurate detection of malicious participants exhibiting on-off and/or data corruption attack, and eventually ensure more accurate data prediction in terms of less data prediction error.

## 6.3 Reputation System

In this section, we describe our reputation and trust distribution mechanisms. We discuss the attack model. Moreover, different components of the trust computation module are discussed in detail.

### 6.3.1 Malicious Entities and Attacker Strategies

No encryption mechanism is applied in the mobile crowdsensing-based IoT application during the data collection and transmission phases. Anyone can participate in the sensor data collection procedure, making it light-weight, and more scalable at lower cost. However, a malicious participant can disrupt the system by launching on-off attack and/or data corruption attack. Due to the absence of an authentication mechanism, a malicious participant can inject false data easily.

#### On-Off Attack

It is a sophisticated attack and harder to detect and prevent. On-off attack means that malicious participants behave good and bad alternatively, hoping that they can remain undetected while hampering data quality [SHYL06], [LGS15]. Most of the state-of-the-art methods fail to detect the on-off attack and thus can not ensure data accuracy in the presence of on-off attackers. In this type of attack scenario, a participant is aware of honest behavior. In other words, s/he knows what the original sensing data of a particular spatiotemporal unit is. They report the correct information for a long time to attain higher reputation value. Then, these malicious participants inject false data similar to high spike to manipulate the sensor data. Their motivation is to change the aggregated data and as a result, resulting into incorrect decision.

#### Data Corruption Attack

Data corruption attack occurs when a participant sends incorrect data either deliberately or recklessly. The reckless error occurs when a participant heedlessly performed the task of sensing or was caused by a sensor error. In contrast, a malicious participant can intentionally fabricate the sensed data to penetrate the system [TPI+18]. For example, in air quality monitoring, a malicious participant may hold the sensor beside a burning cigarette or place it over sand instead of facing to the air. Thus, the reported data will not represent the actual air quality. We considered two types of false data injection rate. First, the malicious entities inject false data in a constant rate throughout the experimental duration. In the second, the rate of false data injection by a malicious user varies at different time instances. However, in the data corruption attack, if a participant is malicious, s/he does not behave as an honest participant any time. The false data injection rate is randomly selected from the range of 30% to 75%.

We assume the devices are properly calibrated before the experimentation, i.e., participating in the sensing task. Thus, calibration error is out of the scope of this article. We focus on the data inaccuracy caused by participants inadvertently or intentionally.

### 6.3.2 RCoD Mechanism

Fig. 6.2 illustrates the main components of our RCoD mechanism. Sensed air quality data are reported by various participants to the server. Afterwards, all these contributions are inputted to the Trust and reputation module. Here the contributions' trustworthiness is analyzed considering different properties (Section 6.3.3 – Section 6.3.5). Each participant's reputation is calculated which reflects

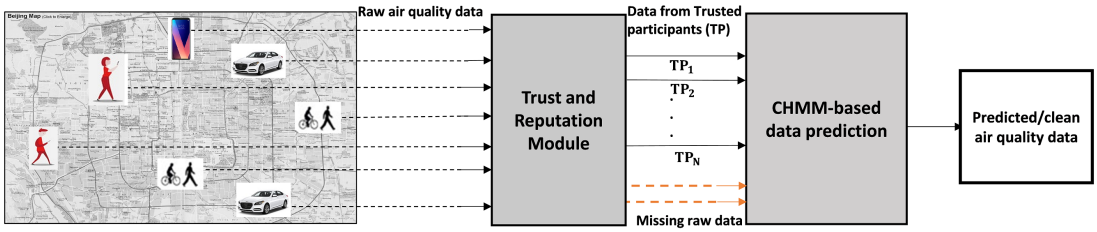



Figure 6.2: Block scheme of RCoD

historical behavior (Section 6.3.6). Based on their reputation, trusted participants are identified. Next, data from trusted participants (with higher reputation scores) identified in the previous step (Section 6.3.6) are input to the contextual hidden Markov model. Also, the data streams containing missing data are taken as input. Finally, we describe the CHMM based data prediction methodology (Section 6.4). This accurate predicted data is used to generate pollution maps as depicted in Fig. 6.1.

### 6.3.3 Dynamic set of Trusted Participants

In most of the state-of-the-art methods, the ground truth value is calculated from the data reported by all the participants. In contrast, we dynamically update the set of trusted participants. The ground truth value is calculated from the data reported by the trusted participants. Since we periodically update this set of trusted participants, it is ensured that the trusted set does not include malicious participants who are trying to forge data. Thus, the data accuracy and robustness of the system is maintained. Our mechanism can detect malicious users and mitigate the false data injected by these users.

In the initialization phase, when there is no historical data, the similarity between contributions received from multiple participants are calculated. If $C_i^k$ is the sensor

data of type $k$ provided by participant $i$, then its similarity with all other data of type k contributed by the other participants regarding the spatiotemporal unit are calculated. The normalized average difference is calculated to be used in the exponential-based initial contribution score generation. The contribution scores range in between 0.36 ($=e^{-1}$) and 1. This score is an input to the reputation table. Then, these initial contribution scores are sorted in descending order. The top $TP$ participants are selected from the sorted list to be declared as the set of trusted participants. The number $TP$ is application dependent. We calculate the ground truth value using Eq. 6.1 from the data reported by participants belonging to the $trustedSet$.

$$Groundtruth = \frac{\sum C_i}{|trustedSet|}, \qquad \forall i \in trustedSet \tag{6.1}$$

Here, i denotes the participant id, $C_i$ is the contribution data provided by participant i.

Now, we briefly describe Algorithm 6: the formation of the initial trusted set. It takes the data contributions made by all the participants on the first day to return an initial set of trusted participants. First, all the participants who contributed data on day 1 are listed. Then the difference between the data reported by different participants at the same epoch (temporal unit) is calculated (lines 6-9). In line 10, the average difference value is calculated for a specific participant i and stored in the array diff (line 10). Next, in line 14, the normalized difference is calculated. For this purpose, the minimum and maximum values of the array $diff$ are identified. Moreover, the reputation for each participant is calculated and stored in the global reputation table. This table contains three columns. The first column contains the participant id, second contains the initial_contribution score, and the final column is dedicated to storing the reputation score. The reputation score is calculated

**Algorithm 6** Formation of Initial Trusted Set

**Input:** day 1 Participant Contributions

1: **Initial Trusted Set**
2: **for** $epoch = 1 : day1\_end$ **do**
3: $participantSet \leftarrow id$
4: $participantLen = |participantSet|$
5: **for** $i = 1 : participantLen$ **do**
6: **for** $j = 1 : participantLen$ **do**
7: $C_{ij} = abs(data_i - data_j)$
8: $Total\_C_{ij} = Total\_C_{ij} + C_{ij}$
9: **end for**
10: $diff(i) = \frac{Total\_C_{ij}}{participantLen}$
11: **end for**
12: **for** $p = 1 : participantLen$ **do**
13: $reputation\_table(p, 1) \leftarrow participantSet[p]$
14: $Norm(p) \leftarrow \frac{diff(p) - min(diff)}{max(diff) - min(diff)}$
15: $reputation\_table(p, 2) \leftarrow e^{-Norm(p)}$
16: $reputation\_table(p, 3) \leftarrow$ Algorithm 8$(p, reputation_table)$
17: **end for**

18: **Select Trusted Participants**
19: $trusted_{no} = \frac{participantLen}{2}$
20: $sortedList = sort(reputaion\_table, descend)$
21: $trustedSet \leftarrow top(reputaion\_table, trusted\_no)$
22: **end for**

by calling Algorithm 8. In lines 13- 16, the reputation table is updated with the calculated information regarding day 1 participants.

Next, in the Selected Trusted Participant method, the trusted set is defined. First, the size of the set is defined as half of the total number of participants (line 19). Then, the participants are sorted in descending order based on their reputation value. Finally, the top pre-defined number of participants are selected and assigned in the trusted set.

Table 6.1: Notations and Their Description

| Notation | Description |
|---|---|
| $pc$ | Participant count |
| $C_i^k$ | Sensor data of type k contributed by participant i |
| $\alpha_i^j$ | Contribution score of participant i with sensor type j |
| $trustedSet$ | Set of trusted participants |
| $\beta_i$ | Proximity score of participant i |
| $Rep(P_f)$ | Reputation of feedback provider |
| $Rep(P_i)$ | Reputation of participant i |
| $Num_{FP}$ | Number of feedback providers |
| $Feed_f(P_i)$ | Feedback from participant f regarding data reported by participant i |
| $\theta_i$ | Feedback score of participant i |
| $\delta_i^j$ | Willingness of participant i for sensor data type j |
| $Context_{trustedSet}$ | Context value of the trusted set of participants |
| $\gamma_i^j$ | context score of participant i with sensor type j |
| $t_{duration}$ | Application-dependent threshold |
| $\lambda_i$ | Timeliness score of participant i |
| $Trust(C_i^j)$ | Trust of participant i with sensor type j |
| $C$ | Context matrix |
| $N$ | Number of participants |
| $Y$ | Reported data matrix |
| $T$ | Final epoch |
| $OV$ | Observation dependence vector |
| $CV$ | Hidden contextual dependence vector |

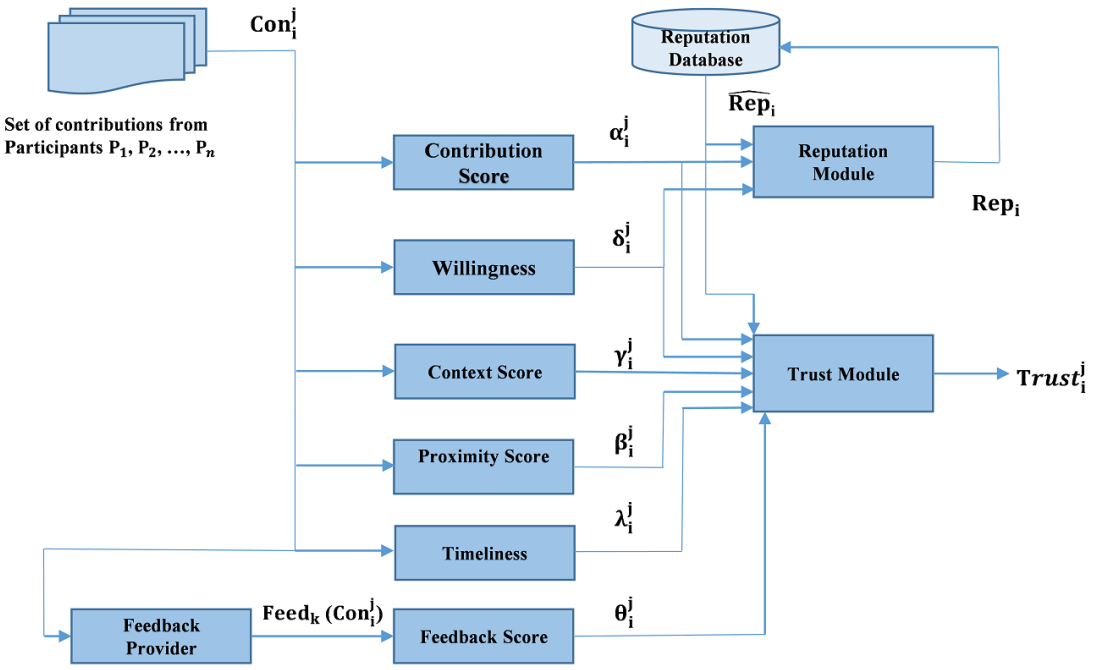



Figure 6.3: System Model

### 6.3.4 Trust Value Assignment

#### Contribution Score

In this section, a score is assigned for the recent data contribution made by a participant. *pc* denotes the participant count, the number of participants who contributed data. The data is compared with the reported data about the same spatiotemporal unit from trusted participants. The sensor data of type $j$ contributed by a participant $i$ is compared against the reports of the same type from the trusted set of participants. The difference values are normalized using Eq. 6.4. Here, $dif_i^j$ is the absolute difference for participant i with sensor data type j. $min(dif_i^j)$ and $max(dif_i^j)$ denote the minimum and maximum difference among all the participants during that epoch. After normalization, the difference values belong to the range $[0, 1]$. The value 0 means the contribution is the same as the trusted participants.

The normalized score is input to the exponential equation Eq. 6.5 to calculate the contribution score ($\alpha_i^j$).

The output value of Eq. 6.5 has the maximum value of 1 and minimum value of $e^{-1}$.

$$data_{trustedSet} = \frac{\sum_{k=1}^{|trustedSet|} data_k}{|trustedSet|} \tag{6.2}$$

$$dif_i^j = abs(data_i^j - data_{trustedSet}), \forall i \in \{1, 2, \ldots, pc\} \tag{6.3}$$

$$Normdif_i^j = \frac{dif_i^j - min(dif_i^j)}{max(dif_i^j) - min(dif_i^j)} \tag{6.4}$$

$$\alpha_i^j = e^{-Normdif_i^j} \tag{6.5}$$

**Proximity Score**

If the sensing data type does not fluctuate much based on distance, is stable throughout a wide spatial area (e.g., grid), then the proximity score is of type Boolean with having value either 0 or 1. If two of the participants' (e.g., $P_i$ and $P_j$) location belongs to the same grid, then they will have similar sensed value. Here i and j have the value from 1 to the total number of participants. The proximity score is calculated using Eq. 6.6.

$$\beta_i = \begin{cases} 1, & grid(P_i) \cap grid(P_j) \neq \phi, \\ 0, & \text{otherwise.} \end{cases} \tag{6.6}$$

On the other hand, some of the applications are location sensitive. The value changes significantly with the increase of the distance between the source and the participant who reports the sensed value. In reality, the nature of the applications

such as diffusion and dispersion play a significant role in the data variation. For example, in pollution detection or noise monitoring applications, a participant locating close to the data source will be able to render the most accurate data of the phenomenon. For these highly location-sensitive sensing applications, we calculate the inverse of the Gompertz equation for assigning proximity score to each data contribution (Eq. 6.9). The L2-norm, calculated using Eq. 6.7, is input to the Eq. 6.8.

$$||L||_2 = \sqrt{(target_x - x_i)^2 + (target_y - y_i)^2} \tag{6.7}$$

$$exponent_i = relevance_b \times e^{-(relevance_c \times ||L||_2)} \tag{6.8}$$

$$\beta_i = 1 - relevance_a \times e^{-exponent_i} \tag{6.9}$$

There are three parameters for the inverse Gompertz function $relevance_a$, $relevance_b$ and $relevance_c$. The parameter $relevance_a$ controls the higher asymptote on the y-axis. The displacement on the x-axis is controlled by the parameter $relevance_b$. The final parameter $relevance_c$ controls the function's decay rate. $(target_x, target_y)$ is the target sensing location and $(x_i, y_i)$ denotes a current location of the participant.

### Rating Score Validation

In a periodic manner, feedback from other participants is gathered for the verification of contributions/sensed data provided by a participant about a particular geographic location. While executing the verification task, the users having a higher reputation score than the target participant are selected. If the data variance is within a tolerable range, then the system assures that the target participant is trustworthy in that particular time instance. In contrast, if most of the higher reputable participants report that the data contributed by the target participant does not match

with the actual sensed value of that spatiotemporal unit, a negative feedback score is assigned to the target participant. Since the aggregated feedback (Eq. 6.10) is considered for assigning the final feedback score, it is resilient against unfair rating attack. An individual cannot successfully disrupt the system's trustworthiness by providing negative feedback to an honest participant. Consequently, the on-off attack of malicious participants is prevented.

$$\theta_i = \frac{\sum_{f=1}^{Num_{FP}} Rep(P_f) \times Feed_f(P_i)}{\sum_{f=1}^{Num_{FP}} Rep(P_f)} \qquad \forall f \in Num_{FP} : Rep(P_f) \geq Rep(P_i) \tag{6.10}$$

Here, $Num_{FP}$ is the number of feedback provided, $P_i$ denotes the participant for whom the feedback is collected. In Eq. 6.10, the reputation value of the feedback provider is used as a weight in the feedback score ($\theta$) calculation. Feedback from a higher reputable participant has a higher influence in the calculation of combined feedback score.

**Willingness**

The number of non-missing data provided by the participant among all the contributions during a certain duration. In the data set, if a participant is located in a spatiotemporal unit but did not report data for consecutive time instances, it means the participant is lacking the willingness to participate. We assume for this work that battery level is not a reason for data inconsistency or missing data. The smartphones used for the sensing purpose had enough energy storage during the experiment.

$$\delta_i^j = \frac{\sum_{t=1}^{cur_{epoch}} |C_i^t \setminus empty(C_i^t)|}{|C_i^t|} \tag{6.11}$$

Here, the ratio of the number of non-empty contributions and total contributions made by participant i of data type j during the previous t epochs is calculated. t has the value from 1 to current epoch.

### Context Score

We compare the similarity of the contextual value of a participant with the context value of the trusted participants who reported the similar type of sensor data at the same time instance. The context value of sensor data type $j$ contributed by a participant $i$ is compared against the context value of the trusted set of participants. The difference values are normalized using Eq. 6.14. After normalization the difference values ($Contextdif_i^j$) belong to the range $[0, 1]$. The value 0 denotes that the context value of the participant is the same as the trusted participants. Then, the normalized score ($ContextNormdif_i^j$) is input to the exponential equation Eq. 6.15 to calculate the context score.

$$Context_{trustedSet} = \frac{\sum_{c=1}^{|trustedSet|} Context_c}{|trustedSet|} \tag{6.12}$$

$$Contextdif_i^j = abs(Context_i^j - Context_{trustedSet}), \forall i \in \{1, 2, \ldots, pc\} \tag{6.13}$$

$$ContextNormdif_i^j = \frac{Contextdif_i^j - min(Contextdif_i^j)}{max(Contextdif_i^j) - min(Contextdif_i^j)} \tag{6.14}$$

$$\gamma_i^j = e^{-ContextNormdif_i^j} \tag{6.15}$$

Where $\gamma_i^j$ denotes the context score of participant i with sensor type j. $pc$ is the count of the participants co-located in the same spatiotemporal unit.

**Timeliness**

This property checks if the participant reported data in a timely manner. The difference between the task assigned ($t_a$) and the data reported ($t_r$) is taken into consideration. If the difference is greater than application-dependent threshold $t_{duration}$, then that data is stale and the timeliness score for the participant will be zero. On the other hand, if the difference is low, it insinuates that the participant carried out the sensing task expeditiously. We calculate the timeliness score ($\lambda$) using the inverse Gompertz function. If $t_a$ is the task assignment time and $t_r$ is the time when the data was reported then,

$$t_{diff} = t_r - t_a \tag{6.16}$$

$$\lambda_i = \begin{cases} a \times e^{-be^{c \times t_{diff}}}, & \mathrm{t}_{diff} \leq t_{duration}, \\ 0, & \text{otherwise.} \end{cases} \tag{6.17}$$

There are three parameters for the inverse Gompertz function (Eq. 6.17 ) 'a', 'b' and 'c'. The parameter 'a' controls the higher asymptote on the y-axis. The displacement on the x-axis is controlled by the parameter 'b'. The final parameter 'c' controls the function's decay rate. When the value of $t_{diff}$ is equal to $t_{duration}$, the timeliness score is almost equal to zero, such as 0.01. We calculate the value of b using the equation.

### 6.3.5 Trust level mapping

The trust level of each contribution made by a participant is calculated. It is a combined metric consisting of the above mentioned six property values: contribution score, proximity score, rating score, context score, willingness, and timeliness. The

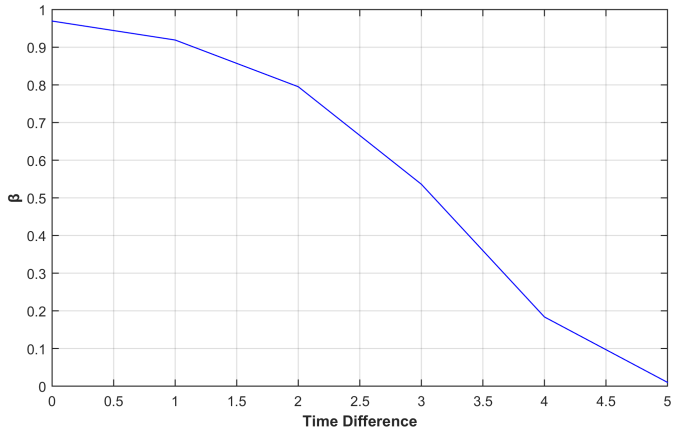



Figure 6.4: Timeliness Score($\lambda$) vs. Time difference using Inverse Gompertz Function

**Algorithm 7** Heterogeneous Trust distribution System

**Input:**Participant Contributions,epoch_begin, epoch_end

**for** $epoch = epoch\_begin : epoch\_end$ **do**
  $P_{List} \leftarrow ID$
  **for all** $P_i \in P_{List}$ **do**
    $\alpha_i^j \leftarrow$ Eq. 6.5
    $reputation\_table[P_i, 2] \leftarrow \alpha_i^j$
    **if** geo-stable application **then**
      $\beta_i \leftarrow$ Eq. 6.6
    **else**
      $\beta_i \leftarrow$ Eq .6.8
    **end if**
    $\theta_i \leftarrow$ Eq. 6.10
    $\gamma_i^j \leftarrow$ Eq. 6.15
    $\lambda \leftarrow$ Eq. 6.17
    $rep \leftarrow reputaion\_table[P_i, 3]$
    $reputation\_table[P_i, 3] \leftarrow$ Algorithm 8$(P_i, reputation_table)$
    $Trust(C_i^j) = w_1 \times \alpha_i^j + w_2 \times \beta_i + w_3 \times \theta_i + w_4 \times \gamma_i^j + w_5 \times \delta_i^j + w_6 \times \lambda_i + w_7 \times rep$
  **end for**
**end for**

reputation value (rep) of the previous epoch is also included in the combined trust level calculation.

$$Trust(C_i^j) = w_1 \times \alpha_i^j + w_2 \times \beta_i + w_3 \times \theta_i + w_4 \times \gamma_i^j + w_5 \times \delta_i^j + w_6 \times \lambda_i + w_7 \times rep \quad (6.18)$$

where $\sum_{i=1}^{7} W_i = 1$. We assign initial trust value of 0 to all participants. As a result, a new participant can not simply inject false data.

The heterogeneous trust distribution method is summarized in Algorithm 7. It takes data contributions made by participants, starting epoch and ending epoch as inputs. It is an iterative process that calculates the trust score for all the participants who reported data in that epoch. The iteration continues from the input value epoch_begin and finishes at the epoch_end. First, all the participants that reported data in the considerable epoch are included in a list. Then, for each participant belonging to the list, the contribution score is calculated using Eq. 6.5 (line 5). In the next step, based on the application type (e.g., location sensitive or location stable) the relevance score ($\beta$) is calculated. For a geo-stable application, the equation used is Eq. 6.6, otherwise Eq. 6.8 is used (lines 5-9) for $\beta$ calculation. The rating score ($\theta$), context score ($\gamma$) and timeliness score ($\lambda$) are calculated calling Eq. 6.10, Eq. 6.15 and Eq. 6.17, respectively (lines 11-13). Next, the reputation table is consulted to obtain the reputation score of the participant. The value assigned to rep is the old reputation value of the participant. To update the reputation table for the current contribution, Algorithm 9 is called in line 15. Finally, the trust value of the contribution made by a participant of type j is calculated in line 16. The total aggregated value of $w_1, w_2, w_3, w_4, w_5, w_6$ and $w_7$ is equal to 1.

### 6.3.6 Reputation Score

The reputation score is dependent on the contribution score and willingness of a participant. It insinuates the historical behavior of a crowdsensing participant.

**Algorithm 8** Reputation computation

**Input :** $Id$, reputation_table
**Output :** Reputation value ($rep$)

1: $dataLen \leftarrow |reputation\_table|$
2: **for** $l = 1 : dataLen$ **do**
3:  **if** $reputation\_table[l, 1] == Id$ **then**
4:  $old_reputation \leftarrow reputation\_table[l, 3]$
5:  $contribution \leftarrow reputation\_table[l, 2]$
6:  **if** $contribution \geq threshold \wedge willingness \geq w_{threshold}$ **then**
7:  $new\_reputation = min(1, (old\_reputation + rewards\_val))$
8:  **else**
9:  $new\_reputaion = max(0, (old\_reputation - punish\_val))$
10:  **end if**
11:  $Reputation\_table[l, 3] = new\_reputation$
12:  **end if**
13: **end for**
14: return $rep$

In the reputation score calculation procedure, a higher punishing score for incorrect contribution than reward score ensures the degradation of the reputation score of a malicious participant trying to vandalize the system performance through exploiting on-off attack. Even if s/he gained high reputation due to showing honest behavior through providing correct sensor data, because of recent incorrect contributions, her/his trust score, as well as the reputation score, will fall below the threshold value. Hence, the participant will not be included in the trusted list for future time instances. When the reputation score falls below the threshold, our system ensures that the participant provides a longer period of correct contributions to regain the honest status back.

Now we briefly describe Algorithm 9: Reputation Computation. It takes as an input participant id, current reputation table and returns the updated reputation value for that participant. The reputation value for each participant is initialized at 0. First, iteratively the participant id is searched in the input reputation table (lines 1-3). The old reputation value, which is obtained from the third column of the

reputation table, corresponding to that participant is recorded in the old_reputation variable. Furthermore, the contribution value is accessed from the second column of the reputation table and compared with the predefined threshold value (lines 5-6). If the value is greater than the threshold, it insinuates the correct contribution made by the participant. Thus, a reward value is added to the old reputation value. However, the reputation value cannot exceed the highest value of 1.0. To ensure this, the minimum of new calculated reputation and 1 is selected as the new_reputation in lines 6-7. In contrast, if the contribution score is below the threshold, it insinuates that the participant reported incorrect data. Thus, a punishment score is applied over the past reputation value. Again, to maintain the minimum reputation score of 0, maximum of zero and calculated reputation is selected and assigned as new_reputation (lines 8-9). Otherwise, due to multiple punishments, a reputation value achieves a negative value. Finally, the reputation table is updated with the new_reputation value.

## 6.4 Reputation-aware Data Prediction Methodology

In this section, the reputation-aware data prediction methodology is discussed. Data from trusted participants identified by the reputation module, discussed in the earlier section, are consulted in the contextual hidden Markov model based missing data prediction methodology.

### Contextual Hidden Markov Model

In the traditional hidden Markov model (HMM), there is only temporal relationship; by adding the contextual layer represented by $c_1, c_2, \ldots, c_N$ we incorporated spatio-temporal and contextual dependence. According to the definition of HMM, there

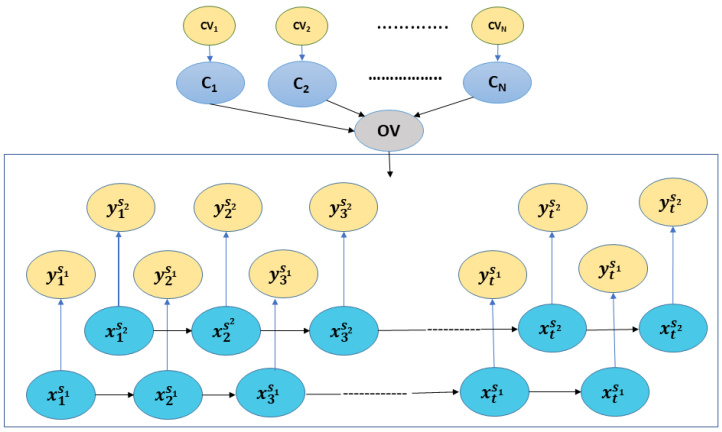



Figure 6.5: Contextual hidden markov model graph diagram

exists a hidden Markov process $x_t$ and an observation $y_t$ is controlled by $x_t$ [Gha01]. Here $s_1, s_2, \ldots, s_N$ denotes different participants who reported data. In Fig. 6.5, the lower diagram enclosed by the rectangle is similar to traditional HMM. The observation value $(y_t)$ is dependent on $x_t$. Here, $x_t$ denotes hidden state at epoch $t$. Epoch is the temporal unit used in this chapter. In Fig. 6.5, each row in the lower enclosed box is dedicated to an individual participant who reported data. For example, the second row from the bottom is dedicated to all the data samples reported by participant id = 2 at different time instances (epoch=$1, 2, \ldots, T$). Here, the final epoch is denoted by $T$. The reported data matrix $Y$ has the dimension of $N \times T$. $N$ is the number of participants who reported data during the time instances commencing at 1 and ending at $T$. On the other hand, the size of the context matrix $C$ is $N \times N$.

We incorporate a new hidden observation dependence vector $OV$ and hidden contextual dependence vector $CV$. For all continuous data streams, the general Markovian dependence $(y_t|x_t)$ is replaced by $(y_t|x_t, OV)$. Here, the observed correlation has a combined influence of $OV$ and $CV$, and thus the context matrix $C$

**Algorithm 9** Missing Value Estimation

**Input :** time, participant_id, TrustedSet

1: $t \in time$ & $s \in participant\_id$
2: **if** data $sen_t^{(s)}$ is available **then**
3: continue
4: **else**
5: $\forall s^{trusted} \in TrustedSet_t$
6: $cm_0 \leftarrow CHMM(sen_t^{(s^{trusted})}, sen_t^{(s)})$
7: **if** $cm_0 = empty$ **then**
8: Predicted_data = Temporal Interpolation
9: **else**
10: Estimated value of $sen_t^{(s)} = cm_0$
11: **end if**
12: **end if**

can be represented using the following conditional probability $p(C|CV, OV)$. The complete likelihood of our proposed CHMM can be noted as,

$$\prod_{t=2}^{T} p(x_t|x_{t-1}) \prod_{t=1}^{T} p(y_t|x_t, OV) \prod_{j=1}^{N} p(C_j|CV_j, OV)p(OV_j)) \tag{6.19}$$

In Eqn. 6.19, the first product depicts the temporal dependence, the second represents observation and the final one is for context.

We calculate the missing data using algorithm 9. It takes epoch, participant id and TrustedSet of participants as input. At first, it is checked if there is any missing data. If the data stream is complete, no further action is required (lines 2-3). Otherwise, the missing data is predicted using the CHMM model. The CHMM is applied to the set of trusted participants and the node that contains missing data. The returned predicted value is assigned to the variable $cm_0$ in line 6. Most of the cases, CHMM is able to return the predicted value. If $cm_0$ is empty, it means there is no correlated sensor that contains data. In that case, temporal interpolation is applied to predict the missing data. Finally, the missing data of the input participant are replaced with the predicted value of $cm_0$.

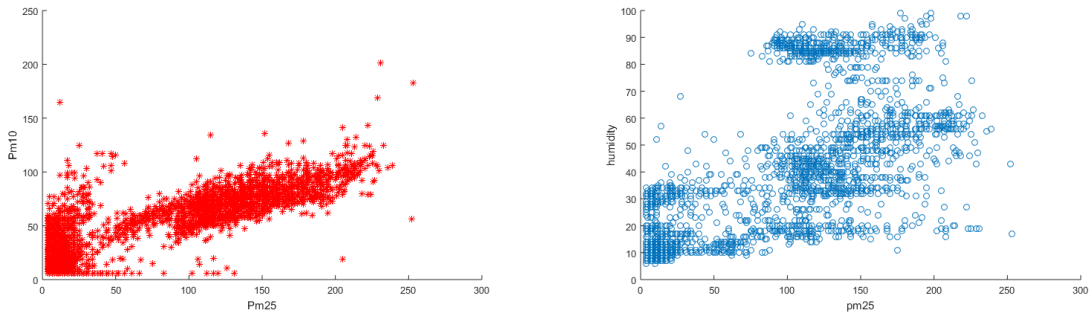
## 6.5 Experimental Evaluation

In this section we describe our experimental setup and environment. We also describe the dataset we used and different performance measures that we considered to evaluate the accuracy of our proposed method. We have used Beijing's air quality data [ZLH13] to implement our algorithm.

### Dataset Description

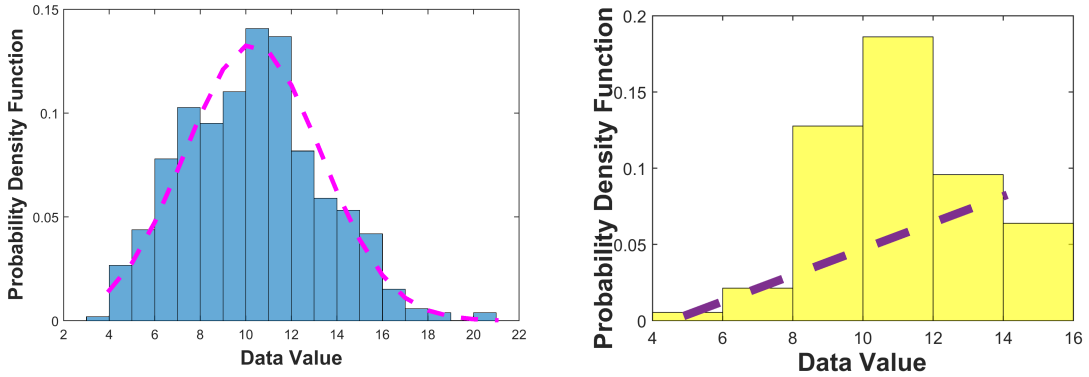
One hundred and forty nine taxis with four types of sensors collect particulate matter with a diameter under $2.5\mu m$ $(PM2.5), PM1.0, NO_2$ and *humidity* data from Beijing during seven days. The air quality data was collected from an area of roughly $120km$ by $150km$. The duration was seven days (149 hours). We assume that the participants are aware of the area where the sensing will take place. Also, the correlation between the different sensors are known. For our experiment, we considered $PM2.5$ as target sensor data type and three types of real correlated sensor data (PM1.0, NO2 and humidity). In Fig. 6.6 (a), the correlation between $PM2.5$ and $PM1.0$ is shown. It can be observed that $PM2.5$ and $PM1.0$ displays linear correlation. In contrast, $PM2.5$ and *humidity* are non-linearly correlated as depicted in Fig. 6.6 (b).

However, the data distribution is not uniform in different areas. In Fig. 6.7 (a), it can be seen that the sensed values follow Gaussian distribution which is denoted by the dashed line. In contrast, the data distribution does not display such regular behavior in other areas as shown in the Fig. 6.7 (b). Our method is dynamic enough to incorporate such heterogeneity.




(a) (b)

Figure 6.6: (a) Correlation of PM2.5 and PM1.0. (b) Correlation of PM2.5 and humidity.




(a) (b)

Figure 6.7: Data Distribution in the area of (a) uniform values, and (b) non-uniform values.

## Simulation Setting

In the original data set, all the participants are honest. Thus, in order to introduce impurity in the sensed data, we incorporate continuous or random errors in the original data. We have considered two types of data impurity: in the first method, the data error percentage for a taxi is unchanged throughout the time period. In the second set of experiments, we consider a random error rate for a single node. We used Python and Matlab programming languages for implementation of our algorithms.

**Assumptions for false data injection**

We considered data corruption and on-off attacks. Similar to the related contemporary works that do not consider colluding among the participants, we had to assume that the majority of the participants are honest for the reputation-based data prediction mechanism. We assumed that malicious users inject false data individually. Unlike most of the state-of-the-art methods, the requirement of the presence of a fixed trusted participant for providing ground truth has been relaxed in our method. Because, the presence of a trusted participant at all time instances is not realistic.

We synthetically injected false data in the original data streams to imitate the false data injection attack. We did two different experiments. In first, the false data injection rate for an individual participant remained unchanged throughout the duration of the experiment. In contrast, the error rate has been varied at different time instances for a participant in the other test cases. For the on-off attack, the false data injection was performed after a long duration of time so that the participant can gain a high reputation value to be considered as a trusted participant. After that, the original data is intermittently replaced with false data to imitate malicious behavior.

We assume the malicious participants do not collude among themselves to infiltrate the data collection procedure. Also, it is assumed that the participants do not perform trial and error attacks, which is a sophisticated attack. In that type of attack, malicious participants can learn different reputation parameters used in the data trustworthiness analysis. Like most of the related contemporary works, we assumed that the malicious participants do not get the chance or in other words do not have enough time to guess the system parameters in order to fool the system. Our method makes data injection attacks harder, but it is breakable by collusion among the malicious participants. In our future work, we will consider collusion

Table 6.2: Parameter Setting

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $w_1$ | 0.4 | $w_5$ | 0.15 |
| $w_2$ | 0.15 | $w_6$ | 0 |
| $w_3$ | 0 | $w_7$ | 0.3 |
| $w_4$ | 0 | threshold | 0.5 |
| reward_val | 0.2 | punish_val | 0.5 |

attacks.

We tested the performance of our algorithm for different levels of erroneous data from malicious users. We also varied the knowledge level of the participants in regards to the experimental environment to imitate sophisticated data manipulation by a malicious crowdsensing participant. In the first test, we assumed the participants do not have any prior knowledge about the experimental environment. The data error ranged from 25% to 75%. One group of malicious participants reported a fixed percentage of error throughout the experiment. This type of error occurs when there is any technical issue in the sensors or the sensor is placed in a covered area during the execution period. In the second experiment, we consider that the malicious participant has extended knowledge about the sensing area. Thus, these participants try to change the sensing data by adding noise to the air quality data of that particular spatio-temporal unit. Moreover, to imitate on-off attack, we applied random data error ranging from 25% to 75% on a participant's reported data stream, after long display of honest behavior. To bring randomness in the behavior, there was no data error in some of the epochs.

## Results and Analysis

We calculated Mean Absolute Error, accuracy, precision, recall, F1 score, AUC, specificity, and Root Mean Square Error for the performance evaluation. We tested

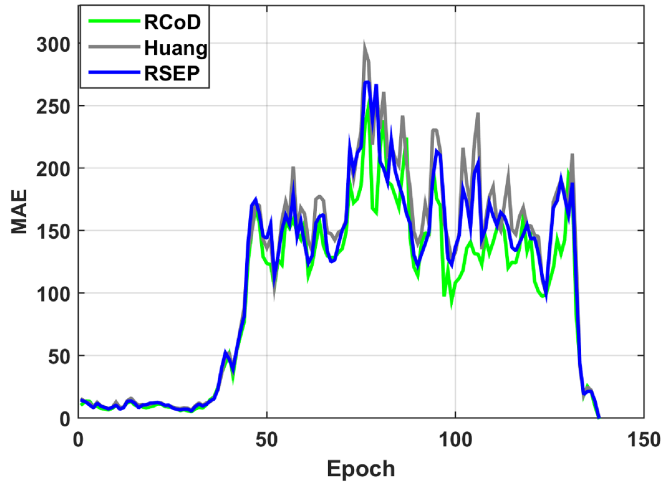



Figure 6.8: Mean Absolute Error Trend in presence of On-off Attack

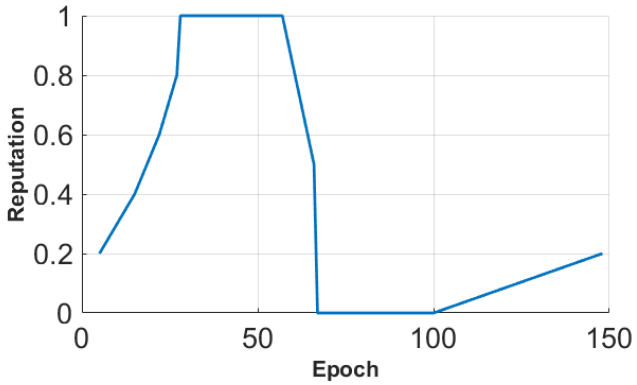



Figure 6.9: Change of Reputation for an on-off attacker

with the presence of different percentages of malicious participants. Most of the state-of-the-art methods assume the presence of malicious participants ranging from 20% to 40%. However, we also tested the scenarios where the majority of the participants are malicious, trying to vandalize the system performance through injecting false data in varied rates.

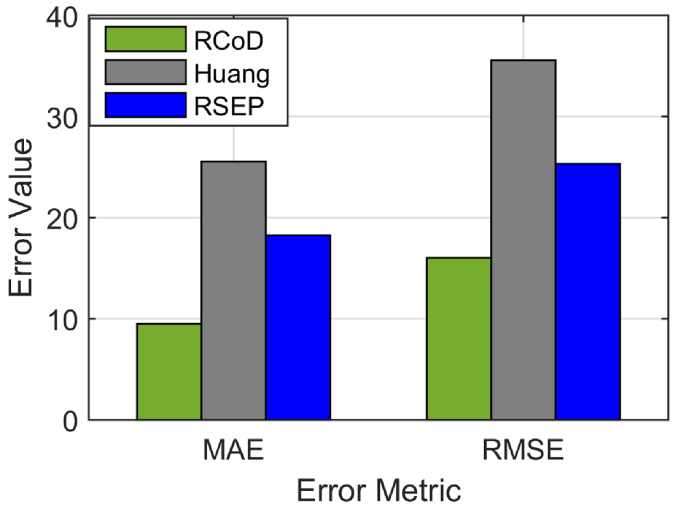



Figure 6.10: Mean Absolute Error and RMSE in presence of On-off Attack

## Mean Absolute Error (MAE)

The mean absolute error is calculated as follows.

$$MAE = abs(V_i - \widehat{V}_i) \tag{6.20}$$

Where $V_i$ is the original value and $\widehat{V}_i$ denotes the predicted value by a method.

In Fig. 6.8, we show the change of Mean Absolute Error (MAE) change per epoch in presence of on-off attack. Our method RCoD incurred less MAE throughout time. When the malicious participants inject false data after achieving the highest reputation (=1.0), the MAE become immense for RSEP [AHZ16] and Huang [HKH14]. The reason behind this is the state-of-the-art methods were not able to detect the data imprecision contributed by the malicious participant who continuously contributed correct data in the past, thus assuming s/he is as an honest participant. The false data injection began from epoch 80. Though at the beginning of the on-off attack, our method incurred high MAE score, causing a sudden spike in Fig. 6.8, RCoD was able to detect the malicious behavior and remove the participant

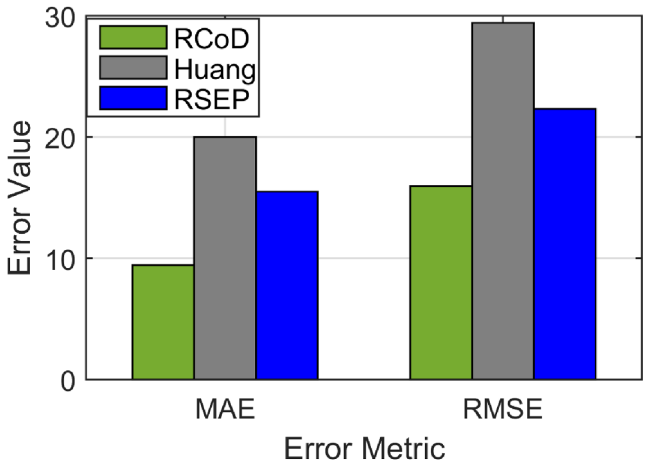



Figure 6.11: Mean Absolute Error and RMSE in presence of 55 Malicious participants

from the trusted participant list. In Fig. 6.9, the change of reputation for an on-off attacker is depicted. It can be observed that the reputation value dropped steeply after attaining the peak value (=1.0). Also, the growth of the reputation is slower than the decay rate. Even after behaving well after around epoch 100, the reputation value did not increase much to be included in the trusted participant list. Moreover, Fig. 6.10 depicts the average MAE value incurred by the methods throughout the seven days. Throughout the seven-day experiment, our RCoD method outperformed RSEP and Huang. Our method achieved 47.98% less MAE than RSEP in the presence of the on-off attack. Also, RCoD incurred 62.82% less MAE than Huang.

We also want to investigate the data corruption attack. RCoD achieved on average 55.45% less MAE than Huang and 48.82% less than RSEP in the presence of varied data corruption attackers. In Fig. 6.12, the prediction performance of our method RCoD, Huang, and RSEP are shown. In this case, the majority of the participants who reported data are malicious, and injected false data at a varied

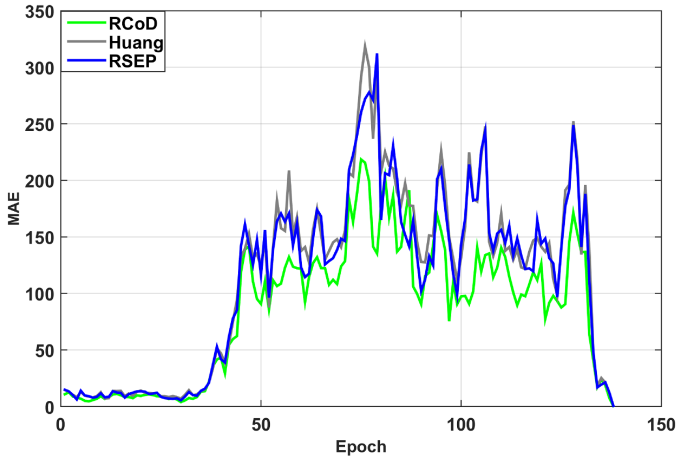



Figure 6.12: MAE Trend for 85 Malicious Nodes

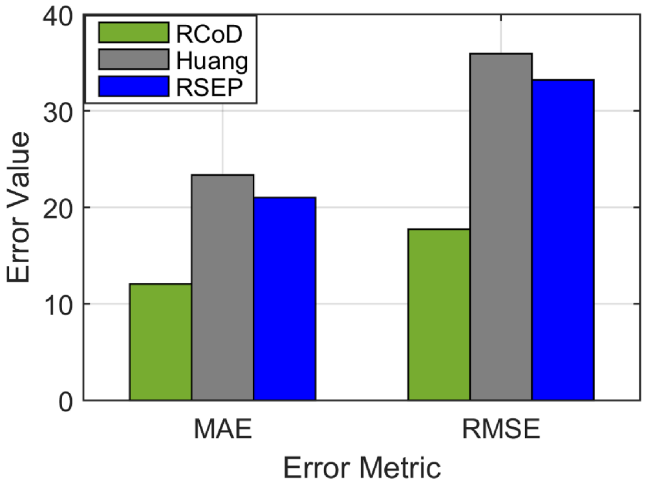



Figure 6.13: Average MAE and RMSE where malicious node is majority

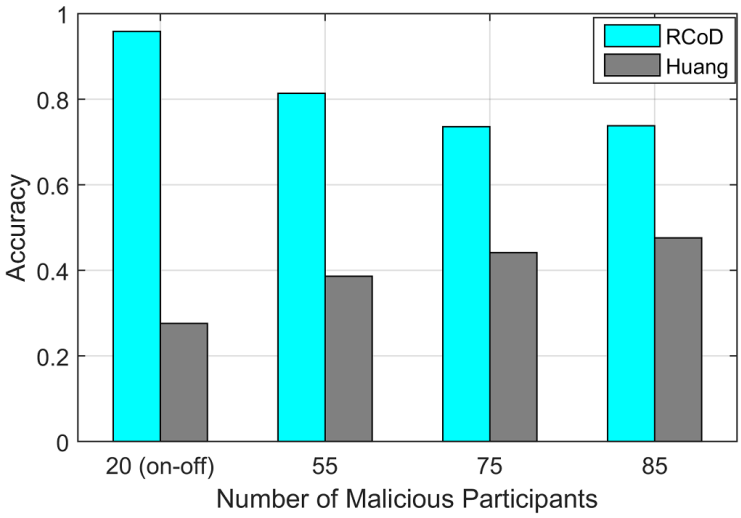



Figure 6.14: Accuracy vs malicious node

error rate ranging from 25% - 75%. This is the worst-case scenario as the number of malicious participants (60%) exceeds the number of honest participants. The state-of-the-art methods fail to predict data accurately in this scenario such as our data prediction performance. The reason behind this is in these works, the majority malicious participants were able to manipulate the fused data in this scenario. On average, from Fig. 6.13 it can be seen that our method incurred 42.61% and 48.33% improvement in terms of less MAE over the RSEP and Huang, respectively.

## Accuracy, Precision, Recall and F1 Score

We measure the performance of our data fusion mechanism quantitatively by calculating the accuracy, precision, recall, and F1 score. Accuracy is the ratio of correct identification of honest or malicious participants among all the detection. A higher value of accuracy insinuates the effectiveness of the method.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad \text{[SL09]} \qquad (6.21)$$

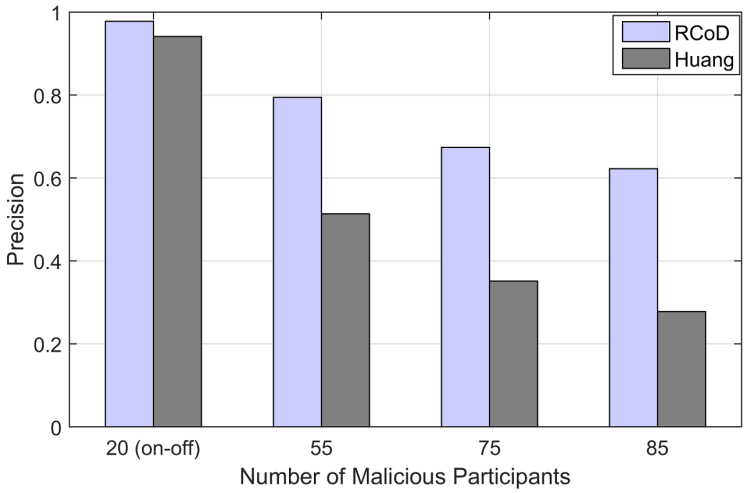



Figure 6.15: Precision vs malicious node

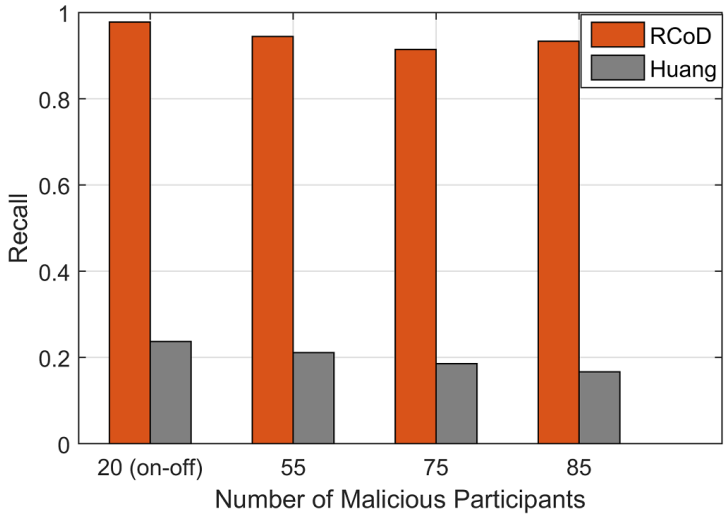



Figure 6.16: Recall vs malicious node

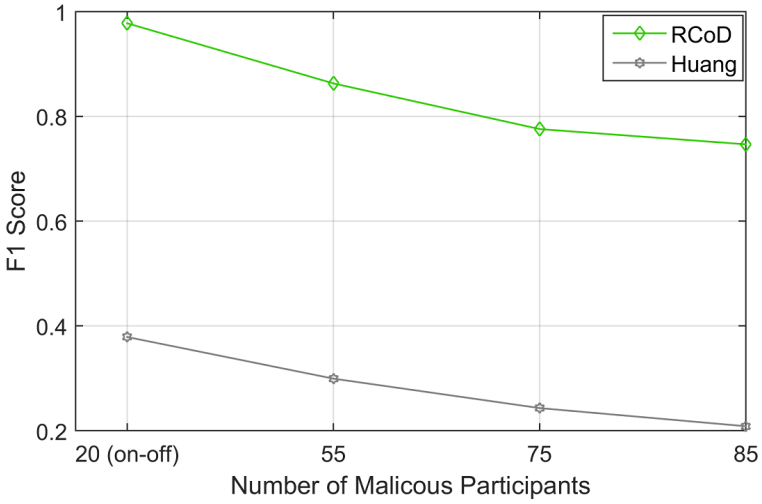



Figure 6.17: F1 Score vs malicious node

Table 6.3: Confusion Matrix for Data Corruption Attack

| Number of Malicious Participants | TP | TN | FP | FN |
|---|---|---|---|---|
| 55 | 85 | 33 | 22 | 5 |
| 75 | 64 | 39 | 31 | 6 |
| 85 | 56 | 51 | 34 | 4 |
| 88 | 19 | 53 | 35 | 38 |

Here TP denotes the number of participants correctly identified as honest, FP denotes the number of participants identified as honest but originally malicious. TN denotes the number of participants identified correctly as malicious and FN is the number of honest participants detected as malicious. Our method was able to receive 74% accuracy in the worst-case scenario (60% malicious). The state-of-the-art methods were not able to predict data accuracy like RCoD since for them the majority malicious participant was able to manipulate the overall data in this scenario. Fig. 6.14 shows that our method achieved high accuracy. In comparison to Huang [HKH14], our RCoD method achieved 49.82% better accuracy on average. The average accuracy value achieved by RCoD is 80%.

$$Precision = \frac{TP}{TP + FP} \qquad \text{[SL09]} \tag{6.22}$$

$$Recall = \frac{TP}{TP + FN} \qquad \text{[SL09]} \tag{6.23}$$

Fig. 6.15 shows the achieved precision values calculated using Eq. 6.22. On average, our method achieved a precision value of 0.77, which is 35.58% higher than Huang. Moreover, in Fig. 6.16 we can see the recall value in the presence of a different number of malicious participants. In the presence of 85 malicious participants, RCoD achieved the recall value of 0.93. The average recall value achieved by our method is 0.94. It represents that our algorithm successfully identified the honest participants with on average 94% cases. Our method outperformed Huang in terms of recall by 78.8%.

Fig. 6.17 shows the F1 score achieved by RCoD and Huang at the presence of a different number of malicious participants. F1 score is calculated using Eqn. 6.24. In the case of on-off attack, RCoD successfully detected the data anomaly, thus achieving F1 Score of 0.98. Furthermore, our method was able to detect the malicious participants with a decent F1 score (=0.75) in the scenario where malicious participants supersede the number of honest participants. In this scenario, F1 score incurred by Huang was only 0.21. Our method achieved at least 61.27% better F1 score than Huang. The average F1 score encountered through out the experiments was 0.84, which indicates the high classification accuracy of our RCoD method.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad \text{[SL09]} \tag{6.24}$$

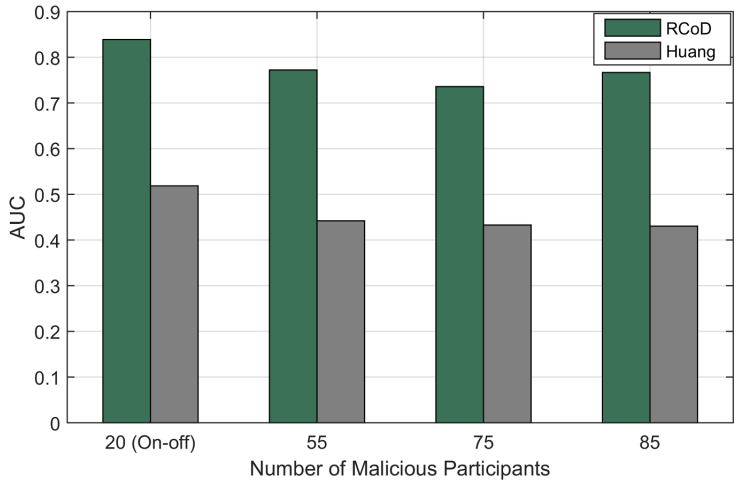



Figure 6.18: AUC vs number of malicious node

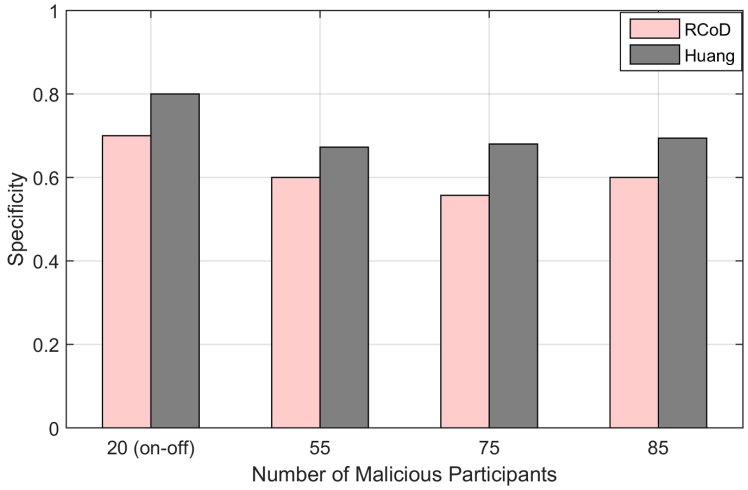



Figure 6.19: Specificity vs number of malicious node

## AUC and Specificity

We calculated AUC and specificity to measure the efficacy of malicious node detection performance of our method. AUC refers to the avoidance of false classification of the classifier. On the contrary, specificity refers to the effectiveness of the method in terms of correct identification of malicious participants [SL09].

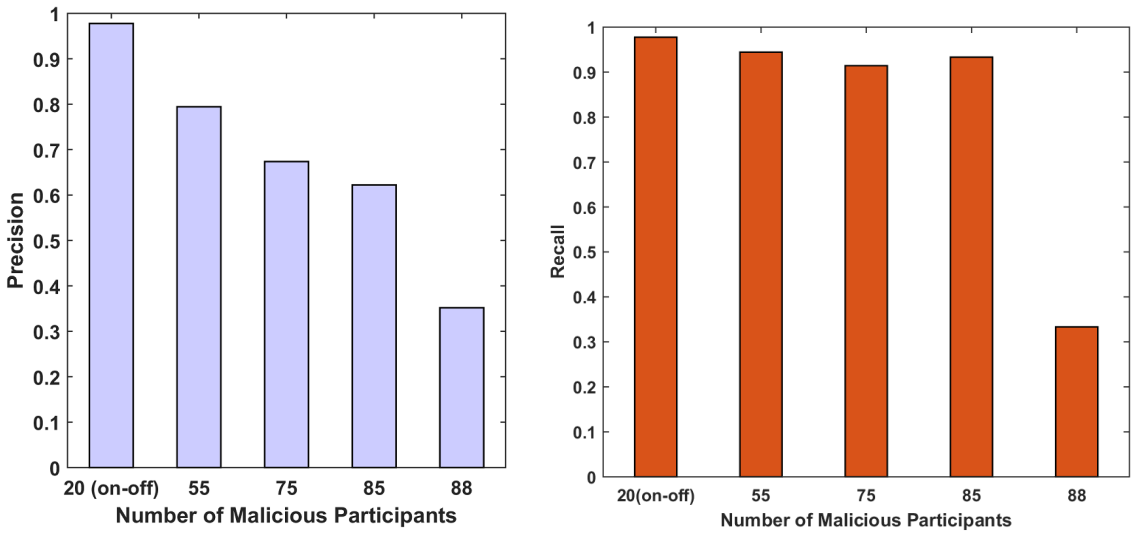
$$AUC = 0.5 \times (\frac{TP}{TP + FN} + \frac{TN}{TN + FP}) \tag{6.25}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6.26}$$

Fig. 6.18 shows the AUC value achieved at the presence of a different number of malicious participants. AUC is calculated using Eqn. 6.25. RCoD outperformed Huang in terms of AUC by 41.5% on average. It can be observed that in the presence of 60% malicious participants, the AUC value is 0.76. It insinuates that our algorithm was successfully able to avoid incorrect classification of participants even where the majority of the participants are malicious. For the on-off attack, the achieved AUC value is 0.84. Furthermore, Fig. 6.19 depicts the achieved specificity values. Even though our main goal is proper identification of honest participants for ensuring the data accuracy of the overall system, the algorithm could identify the malicious participants accurately with an average specificity value of 0.61. In the presence of on-off attacker, our method could properly identify the malicious participants in 70% of cases.

### Breaking point of RCoD

We experimented in the presence of 88 malicious participants among the total of 145 participants to show the breaking point of our method. Here the number of




(a) (b)

Figure 6.20: (a) Precision value and (b) Recall value incurred by RCoD for various number of malicious participants.

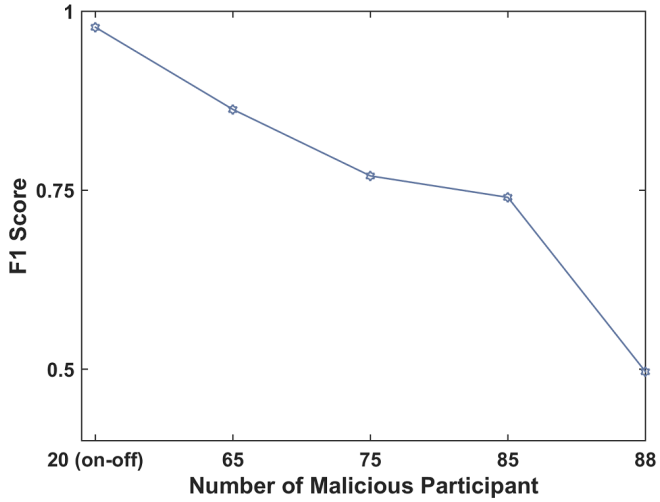



Figure 6.21: F1 Score incurred by RCoD for various number of malicious participants

Table 6.4: Performance Metric in the presence of 88 (greater than 60%) malicious participants

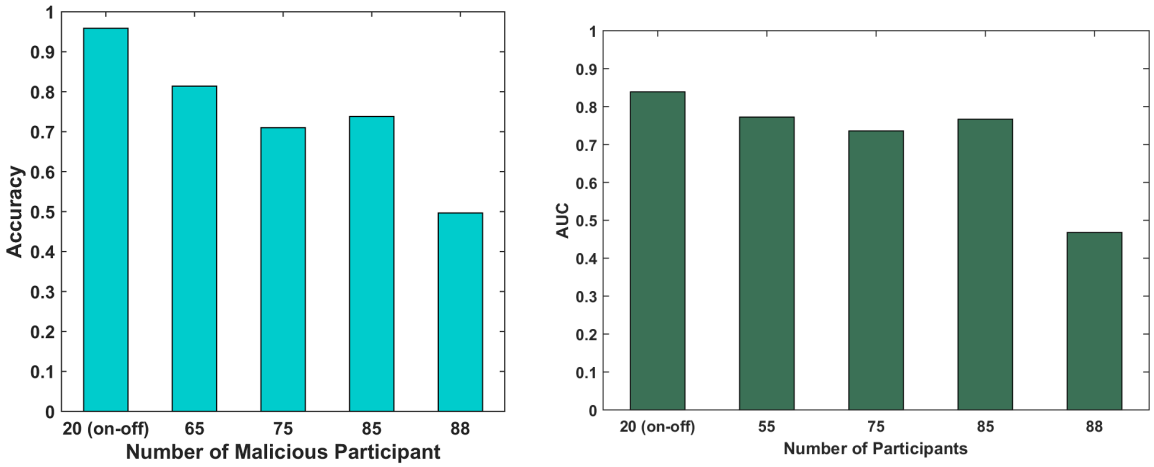
| Precision | Recall | Accuracy | F1 Score |
|---|---|---|---|
| 0.3518 | 0.3333 | 0.4965 | 0.3423 |

Table 6.5: AUC and Specificity in the presence of 88 (greater than 60%) malicious participants

| AUC | Specificity |
|---|---|
| 0.4678 | 0.6022 |

malicious participants is over 60% of the total participants. Table 6.4 and Table 6.5 displays different performance metrics (e.g., Precision, Recall, Accuracy, F1 Score, AUC, and Specificity) in the presence of 88 malicious participants. The performance degrades a lot in comparison to the presence of less than 60% malicious participants. Our Reputation-based context-aware data fusion (RCoD) method fails to identify properly the honest participants when the number of malicious participants is 88. The F1 Score, a measure of classification accuracy is only 0.34 (Fig. 6.22). From Table 6.3, we can see that our method was able to identify 19 honest participants accurately (TP) among the 57 original honest participants.

From Fig. 6.20, we can observe the degradation of the precision score in the presence of 88 malicious participants in comparison to the presence of 85 malicious participants. Similarly, it is noticeable that the recall value reduced 64.34% when the number of malicious participants increased from 60% to 61%. Moreover, Fig. 6.21 depicts the F1 score values incurred in the presence of a different number of malicious participants. We can observe a sharp decline in the F1 score in the presence of 88 malicious participants in contrast to 85 participants. This insinuates that our method failed to classify honest and malicious participants properly when the number of malicious exceed 60% of the total participants. Similar observation can be seen in Fig. 6.22 (b). With 1% increase over 60% malicious participants, the




(a) (b)

Figure 6.22: (a) Accuracy value incurred by RCoD for various number of malicious participants. (b) AUC value for various number of malicious participants.

performance degrades significantly. Thus, our RCoD is resilient against up to 60% of malicious participants.

## Root Mean Square Error (RMSE)

We calculated Root mean square error (RMSE) as a prediction metric. It insinuates the prediction error of a method. $RMSE$ is a standard metric to evaluate the accuracy of the prediction model [ZSS14].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{V}_i - V_i)^2} \qquad \text{[ZSS14]} \tag{6.27}$$

where $\widehat{V}_i$ is the predicted value, $V_i$ is the original value and $n$ is the number of epochs. In Fig. 6.10, we show the RMSE incurred by our RCoD and RMSE method in case of on-off attack. Fig. 6.10 and Fig. 6.13 depict the RMSE value incurred by the methods (RCoD, Huang, and RSEP) where 55 and 85 malicious attackers among the total 145 participants performing data corruption attack. Our method

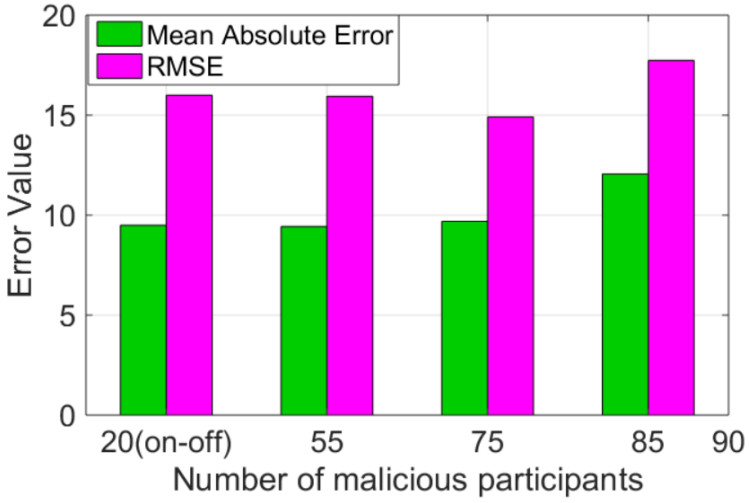



Figure 6.23: Average MAE and RMSE vs malicious node

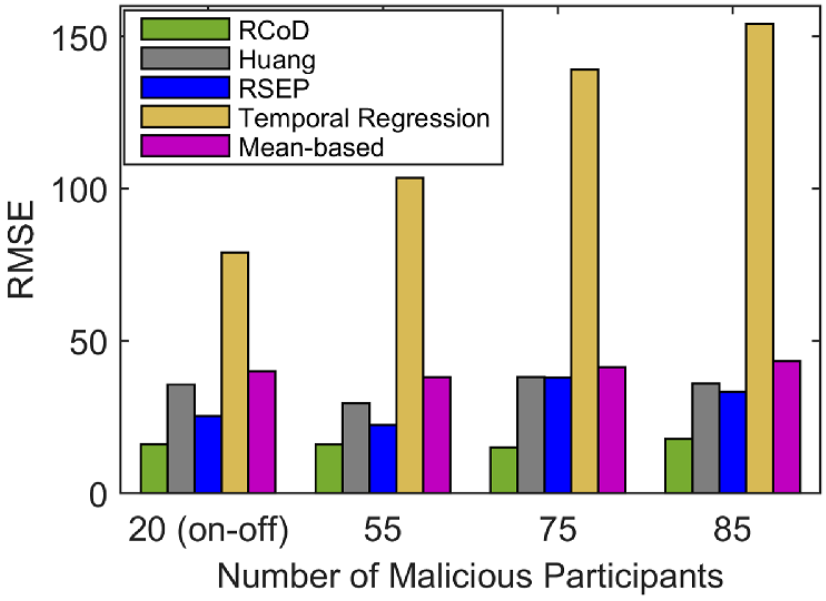



Figure 6.24: Average RMSE vs malicious node

outperformed Huang and RSEP by 45.81% and 28.6% respectively in the presence of 55 malicious participants. On the other hand, in the worst-case scenario where the majority of the participants are malicious, the performance of RCoD is noteworthy. Our method outperformed the closest competitor RSEP by 46.58%. Furthermore, our method incurred 50.62% less RMSE than Huang. We make similar observations for RCoD under a varied number of malicious participants infiltrating the system, as shown in Fig. 6.23 the average prediction error incurred by our method is reasonably low.

Next, Fig. 6.24 presents the RMSE value incurred by our RCoD, Huang[HKH14], RSEP [AHZ16], mean-based [JAF+06], [SGG10] and temporal regression [ŽH13] for on-off attack and data corruption attack. We tested against a different number of malicious participants. Our method outperformed the closest competitor RSEP by incurring 43.15% less RMSE on average. In the presence of varied malicious participants, RCOD incurred at least 45.8% and at most 60.88% less RMSE than Huang. The performance superiority over Temporal regression and mean-based is noteworthy. The reason behind the poor performance of other methods can be explained as they do not incorporate feedback from reputable participants to validate the reported data. Also, our dynamic trusted set and the distribution of contribution score highly reflect the ground truth data. Furthermore, the contextual hidden Markov model exploits the contextual relationship among the participants in the data prediction method. Hence, our RCoD method achieves better prediction than the compared state-of-the-art methods in the presence of a high number of malicious participants.

## 6.6 Discussion and Summary

In this chapter, we developed an online method for data quality prediction in mobile crowdsensing for IoT that considers the spatial, temporal and context relationship among participants. We implemented our methods on Beijing air quality dataset. Most of the state-of-the-art methods assume the presence of malicious participants ranging from 20% to 50%. However, we also tested the scenarios where the majority of the participants are malicious, trying to vandalize the system performance by injecting false data in varied rates. We have tested with the presence of different high numbers (55, 65, 75 and 85 out of 145) of participants injecting false data. Our method was able to receive 74% accuracy in the worst-case scenario (60% malicious participants), exhibiting the quality of resilience. In this case, our RCoD outperformed the closest competitors RSEP and Huang by incurring on average 43.15% and 53.08% less RMSE, respectively. The state- of-the-art methods were not able to achieve prediction data accuracy like our RCoD since for them the majority of malicious participants were able to manipulate the overall data in this scenario. The success of our approach lies in the integration of dynamic trust evaluation of the sensed data that allows us to defend data corruption and on-off attacks, as well as identify malicious or honest participants based on their reported data in real time.

CHAPTER 7

# FUTURE WORK AND CONCLUSION

This dissertation described different methods for real-time data mining in mobile IoT. In this chapter, we present our concluding remarks and provide directions for future work.

## 7.1 Discussions

### Context-aware data cleaning

In chapter 3, we presented a novel mechanism for cleaning environmental sensing data streams that consider not only the sensed value, but also the sensor context and movement affinity for data cleaning. Our proposed method predicted the missing data value with greater accuracy, even when there was a higher fluctuation in the data streams. The average RMSE incurred by our method was only 3.9 at varied data impurity rates. Simulation results showed up to 24% reduction in root mean square error (RMSE) over IMC [ZSS14] and up to 30% compared to mean-based cleaning [JAF$^+$06, SGG10] during the considered time period. We evaluated the approaches based on Smart City project in Melbourne factual dataset.

Although the performance of our context-aware data cleaning method outperformed the related contemporary (IMC [ZSS14] and mean-based [JAF$^+$06, SGG10]) works by achieving less root mean square error in the data prediction, several problems still need to be addressed. First, the scalability test was not performed. The geographic context depends on a priori knowledge of subarea boundaries. Heterogeneous sensor context and the effect of varied semantics in mobile sensing was not considered. Also, we assumed all the nodes to be cooperative in this work. We plan to consider behavior of malicious nodes in the near future.

## Semantic-aware trajectory data mining

In chapter 4, we presented a semantic-aware online clustering-based method for movement relationships finding in mobile trajectories. We incorporated semantics annotation in the raw trajectory data in order to discover various movement relationships between sub-trajectories of mobile devices, and we conducted experiments on a real-world data set. Along with the added advantage of semantic-aware movement behavior analysis, our method identified outliers in the clustering process with almost similar performance (average recall 0.92 and F1 score of 0.8) as classic density-based clustering algorithm DBSCAN [EKS+96].

One direction in which to improve this work is to add different application domain knowledge in the mobile trajectory data. Moreover, we considered stop and go, flock, and moving together movement behavior among the large trajectory data. There are additional types of movement patterns that can be identified.

## Correlated Sensor-based data fusion

In chapter 5, we presented a correlated sensor-based data fusion mechanism to ensure data integrity. The data fusion method is applied to multiple types of sensor data streams. These varied sensors are correlated with each other. The benefits of our proposed mechanism are two-fold: it is capable of defending against a data corruption attack by identifying honest and malicious participants based on their reported data in real time, and it yields fewer data prediction errors. We showed that the proposed mechanism outperformed the existing mean-based [JAF+06, SGG10] and temporal linear regression-based data prediction [ZLH13] models. We evaluated the approaches based on two datasets: Rome crowdsensing temperature and Beijing Air quality datasets. In case of the air quality dataset, our method incurred on

average 25% and 59% less RMSE than mean-based and temporal linear regression models, respectively.

Although the performance of our correlated sensor-based data fusion method outperformed the related contemporary (mean-based [JAF+06, SGG10] and temporal linear regression-based data prediction [ZLH13]) in terms of incurring less root mean square error and percentage error, there remains opportunity for possible improvements. First, if the malicious users show on-off attack behavior, behave well initially and then begin false data injection, the method will not be able to identify that fluctuation, which will in turn degrade data prediction accuracy. Also, it is assumed the location information is correct, and no contextual information has been considered in the data fusion mechanism. Additionally, the geographic context depends on a priori knowledge of sub-area boundaries.

## Reputation-based context aware data fusion

In Chapter 6, we presented a reputation-based context-aware data fusion mechanism. This method is resilient against on-off and data corruption attack behavior of malicious participants; even in the case of high false data injection, our method was able to ensure data reliability. We are the first to use contextual hidden Markov model for online data stream cleaning in the mobile IoT, which facilitated the lightweight data prediction method. In this work, we considered the spatial, temporal and contextual relationships among users while ensuring data quality accuracy. We also considered different spatial granularity scenarios while defining the correlated participants for data fusion based on the application type: spatially stable or variable, which was not considered in the prior research. Our method exhibited its resilience by achieving 74% accuracy in the worst-case scenario (60% malicious

users). In contrast, Huang [HKH14] achieved 39.48% accuracy on average. In the worst-case scenario, our RCoD outperformed the closest competitor RSEP [AHZ16] by incurring 46.58% less RMSE on average. Also, the improvement over Huang was 50.62%.

One direction to improve this work is to incorporate the scenario of a highly sparse network. How the data can be predicted in such a remote geographic location is a challenging problem for future research. We assumed the malicious participants do not collude among themselves to infiltrate the data collection procedure. Also, it is assumed that the participants do not perform trial and error attacks, which is a sophisticated attack. In that type of attack, malicious participants can learn different reputation parameters used in the data trustworthiness analysis. Like most of the related contemporary works, we assumed that the malicious participants do not get the chance or in other words do not have enough time to guess the system parameters in order to fool the system. Our method makes data injection attacks harder, but it is breakable by collusion among the malicious participants. In our future work, we will consider collusion attacks. Also, it is assumed the location data is accurate. Working on imprecise location information is a possible research area to explore.

## 7.2 Future Directions

### Data Prediction in highly Sparse Networks

This research direction inspires the development of a flexible data stream cleaning method that will work efficiently in both dense and sparse networks, where the concurrent presence of more than one sensor in close proximity is not ensured. In

mobile IoT networks, it is not always possible to ensure the presence of a node for sensing. We propose to include the location-aware data offloading to mobile cloud, which is an energy-aware space efficient mechanism, where some data will be offloaded to the nearest mobile cloud. We hypothesize that using mobile cloud will help in providing data for the location about which there is no recent data. Our preliminary work [TCA+14] compares the location-aware code offloading mechanism with CloneCloud [CIM+11] and outperforms it. If there is no sensor present at a point of interest (PoI), which is an example of a sparse network, the mobile cloud will be consulted to receive data about the queried PoI. If it is not found there, then temporal interpolation will be applied to predict the missing data.

### Context-aware Data mining with Location Data Uncertainty

This research direction inspires the development of a data mining mechanism that fuses semantic and contextual information in the identification of various trajectory relationships. Most of the state-of-the-art works assume that the trajectory information received for data mining is accurate. However, imprecise location data is reported by the mobile IoT devices along with the sensed data in real-life applications; as a result, there should be a provision of tolerating this kind of location imprecision in the data mining algorithms.

Generally, when a participant moves to an indoor location or an urban canyon, there is a high chance of losing GPS data, resulting in noisy trajectory data. The uncertainty of the trajectory data prevents an accurate classification of the trajectory relations and identification of trajectory outliers in real time. Hence, there exists an immense need for developing a robust online semantic-aware methodology that endures uncertainty in the trajectory data for estimating noisy sensor locations.

## 7.3 Conclusion

According to the Economist, "The world's most valuable resource is no longer oil, but data" [Eco17]. Thus, ensuring data trustworthiness is highly essential to the IoT applications for providing a decent quality of service. This dissertation is motivated by the immense demand for developing real-time data fusion mechanisms for improving data quality in mobile IoT. In this dissertation, we present online methods for data prediction by fusing spatiotemporal and contextual relationships among the participating resource-constrained mobile IoT devices. In our first study, we propose a data prediction method for missing data streams that considers the spatiotemporal and contextual relationship among the sensors. Next, we develop a real-time outlier detection mechanism analyzing the behavioral movement similarity. Unlike the existing approaches that consider only the spatiotemporal relationship, we incorporate the location semantics which facilitates various movement and behavioral patterns identification in mobile trajectory data. In our earlier works, we assume the sensors to be cooperative, which we relax in the next section. We study the data quality problem in mobile crowdsensing-based IoT systems as well. With various advantages of mobile crowdsensing (e.g., scalability at lower cost), its people-centric architecture introduces more inaccurate data. Due to its open structure, it allows malicious users to interrupt a system by reporting fabricated or erroneous data, making trust evaluation an important issue in these applications. To ensure data integrity in mobile IoT, we develop a reputation-aware context-based data fusion mechanism for data quality prediction. Our online method considers the spatiotemporal, inter-sensor categorization and context relationship among the participants. This method is capable of defending against false data injection by differentiating malicious and honest participants based on their reported data in real time, and

yields a lower data prediction error rate. We hope the future applications of mobile IoT will benefit from our research outcomes, and our study will inspire the development of better data mining solutions in terms of reducing data prediction errors.

# BIBLIOGRAPHY


[AAB+14] F Antonelli, M Azzi, M Balduini, D DellAglio, G Caviglia, P Ciuccarelli, ED Valle, and R Larcher. Towards city data fusion: A big data infrastructure to sense the pulse of a city in real-time. In *European Data Forum 2014*, pages 19–20, 2014.

[ACFM14] Annalisa Appice, Anna Ciampi, Fabio Fumarola, and Donato Malerba. Sensor networks and data streams: Basics. In *Data Mining Techniques in Sensor Networks*, pages 1–8. Springer, 2014.

[AEGPS10] Nils Aschenbruck, Raphael Ernst, Elmar Gerhards-Padilla, and Matthias Schwamborn. Bonnmotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd international ICST conference on simulation tools and techniques*, page 51. ICST (Institute for Computer Sciences, Social-Informatics and ..., 2010.

[AHM+09] Paul M Aoki, RJ Honicky, Alan Mainwaring, Chris Myers, Eric Paulos, Sushmita Subramanian, and Allison Woodruff. A vehicle for research: using street sweepers to explore the landscape of environmental community action. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 375–384. ACM, 2009.

[AHZ16] Mohannad A Alswailim, Hossam S Hassanein, and Mohammad Zulkernine. A reputation system to evaluate participants for participatory sensing. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[AL17] Amany Alnahdi and Shih-Hsi Liu. Mobile internet of things (miot) and its applications for smart environments: A positional overview. In *2017 IEEE International Congress on Internet of Things (ICIOT)*, pages 151–154. IEEE, 2017.

[AS16] Bushra Anjum and Chaman Lal Sabharwal. Filtering compromised environment sensors using autoregressive hidden markov model. *Polibits*, (54):5–10, 2016.

[BBL+14] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from url http://crawdad.org/roma/taxi/20140717, July 2014.

[BD12] Sabbas Burdakis and Antonios Deligiannakis. Detecting outliers in sensor networks using the geometric approach. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1108–1119. IEEE, 2012.

[BGS+13] Joel W Branch, Chris Giannella, Boleslaw Szymanski, Ran Wolff, and Hillol Kargupta. In-network outlier detection in wireless sensor networks. *Knowledge and information systems*, 34(1):23–54, 2013.

[BIT+17] Hedde HWJ Bosman, Giovanni Iacca, Arturo Tejada, Heinrich J Wörtche, and Antonio Liotta. Spatial anomaly detection in sensor networks using neighborhood information. *Information Fusion*, 33:41–56, 2017.

[BL+94] Vic Barnett, Toby Lewis, et al. *Outliers in statistical data*, volume 3. Wiley New York, 1994.

[CB97] Matthew S Crouse and Richard G Baraniuk. Contextual hidden markov models for wavelet-domain signal processing. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No. 97CB36136)*, volume 1, pages 95–100. IEEE, 1997.

[CBD02] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[CDL+19] Ling Chen, Yifang Ding, Dandan Lyu, Xiaoze Liu, and Hanyu Long. Deep multi-task learning based urban air quality index modelling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):2, 2019.

[CDW+15] Feng Chen, Pan Deng, Jiafu Wan, Daqiang Zhang, Athanasios V Vasilakos, and Xiaohui Rong. Data mining for the internet of things: literature review and challenges. *International Journal of Distributed Sensor Networks*, 11(8):431047, 2015.

[CFK+17a] Giuseppe Cacciatore, Claudio Fiandrino, Dzmitry Kliazovich, Fabrizio Granelli, and Pascal Bouvry. Cost analysis of smart lighting solutions for smart cities. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[CFK+17b] Andrea Capponi, Claudio Fiandrino, Dzmitry Kliazovich, Pascal Bouvry, and Stefano Giordano. A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures. *IEEE Transactions on Sustainable Computing*, 2(1):3–16, 2017.

[CFKB17] Andrea Capponi, Claudio Fiandrino, Dzmitry Kliazovich, and Pascal Bouvry. Energy efficient data collection in opportunistic mobile crowd-sensing architectures for smart cities. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 307–312. IEEE, 2017.

[CGF+17] Stefano Chessa, Michele Girolami, Luca Foschini, Raffaele Ianniello, Antonio Corradi, and Paolo Bellavista. Mobile crowd sensing management with the participact living lab. *Pervasive and Mobile Computing*, 38:200–214, 2017.

[CHZT19] Yun Cheng, Xiaoxi He, Zimu Zhou, and Lothar Thiele. Ict: In-field calibration transfer for air quality sensor deployments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):6, 2019.

[CIM+11] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314. ACM, 2011.

[CLK+13] Guangshuo Chen, Xiao-Yang Liu, Linghe Kong, Jia-Liang Lu, Yu Gu, Wei Shu, and Min-You Wu. Multiple attributes-based data recovery in wireless sensor networks. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 103–108. IEEE, 2013.

[CoM18] Australia City of Melbourne. Melbourne data. https://data.melbourne.vic.gov.au/Environment/Trial-Environmental-sensor-readings/ez6b-syvw, October 2018.

[CSK+14] Swarup Chandra, Justin Sahs, Latifur Khan, Bhavani Thuraisingham, and Charu Aggarwal. Stream mining using statistical relational learning. In *2014 IEEE International Conference on Data Mining*, pages 743–748. IEEE, 2014.

[CZVZ14] Angelo Cenedese, Andrea Zanella, Lorenzo Vangelista, and Michele Zorzi. Padova smart city: An urban internet of things experimen-

tation. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–6. IEEE, 2014.

[DKV+09] Antonios Deligiannakis, Yannis Kotidis, Vasilis Vassalos, Vassilis Stoumpos, and Alex Delis. Another outlier bites the dust: Computing meaningful aggregates in sensor networks. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 988–999. IEEE, 2009.

[DSS12] Xin Luna Dong, Barna Saha, and Divesh Srivastava. Less is more: Selecting sources wisely for integration. In *Proceedings of the VLDB Endowment*, volume 6, pages 37–48. VLDB Endowment, 2012.

[dSZdM16] Ticiana L Coelho da Silva, Karine Zeitouni, and José AF de Macêdo. Online clustering of trajectory data stream. In *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, volume 1, pages 112–121. IEEE, 2016.

[Eco17] The Economist. https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data, May 2017.

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[Eln03] Eiman Elnahrawy. Research directions in sensor data streams: solutions and challenges. *Rutgers University, Tech. Rep. DCIS-TR-527*, 2:D3, 2003.

[EN03] Eiman Elnahrawy and Badri Nath. Cleaning and querying noisy sensors. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 78–87. ACM, 2003.

[FCC+17] Claudio Fiandrino, Andrea Capponi, Giuseppe Cacciatore, Dzmitry Kliazovich, Ulrich Sorger, Pascal Bouvry, Burak Kantarci, Fabrizio Granelli, and Stefano Giordano. Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments. *IEEE Access*, 5:3490–3503, 2017.

[FG09] Conny Franke and Michael Gertz. Orden: Outlier region detection and exploration in sensor networks. In *Proceedings of the 2009 ACM SIG-*

*MOD International Conference on Management of data*, pages 1075–1078. ACM, 2009.

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

[Gha01] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. In *Hidden Markov models: applications in computer vision*, pages 9–41. World Scientific, 2001.

[GKD+10] Nikos Giatrakos, Yannis Kotidis, Antonios Deligiannakis, Vasilis Vassalos, and Yannis Theodoridis. Taco: tunable approximate computation of outliers in wireless sensor networks. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 279–290. ACM, 2010.

[GL15] Saul Gill and Brian Lee. A framework for distributed cleaning of data streams. *Procedia Computer Science*, 52:1186–1191, 2015.

[GLN15] Saul Gill, Brian Lee, and Euclides Neto. Context aware model-based cleaning of data streams. In *Signals and Systems Conference (ISSC), 2015 26th Irish*, pages 1–6. IEEE, 2015.

[goo19] google. https://developers.google.com/maps/documentation/javascript/examples/geocoding-reverse, February 2019.

[GYT10] D Deniz Genc, Canan Yesilyurt, and Gurdal Tuncel. Air pollution forecasting in ankara, turkey using air pollution index and its relation to assimilative capacity of the atmosphere. *Environmental monitoring and assessment*, 166(1-4):11–27, 2010.

[Hed19] Andrew Hedges. Haversine. http://andrew.hedges.name/experiments/haversine/, January 2019.

[HKH14] Kuan Lun Huang, Salil S Kanhere, and Wen Hu. On the need for a reputation system in mobile phone based sensing. *Ad Hoc Networks*, 12:130–149, 2014.

[HPK11] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[HXL+17] Xiaohong Hao, Liwen Xu, Nicholas D Lane, Xin Liu, and Thomas Moscibroda. Density-aware compressive crowdsensing. In *Information Processing in Sensor Networks (IPSN), 2017 16th ACM/IEEE International Conference on*, pages 29–40. IEEE, 2017.

[IMR14] Md Zahidul Islam, Quazi Mamun, and Md Geaur Rahman. Data cleansing during data collection from wireless sensor networks. 2014.

[JAF+05] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. A pipelined framework for online cleaning of sensor data streams. *Computer Science*, 2005.

[JAF+06] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. Declarative support for sensor data cleaning. *Lecture Notes in Computer Science*, 3968:83–100, 2006.

[JGB14] Antonio J Jara, Dominique Genoud, and Yann Bocchi. Short paper: Sensors data fusion for smart cities with knime: A real experience in the smartsantander testbed. In *Internet of things (wf-iot), 2014 ieee world forum on*, pages 173–174. IEEE, 2014.

[JGMP14] Jiong Jin, Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami. An information framework for creating a smart city through internet of things. *IEEE Internet of Things journal*, 1(2):112–121, 2014.

[JJLB12] O. Saukh D. Hasenfratz J. J. Li, B. Faltings and J. Beutel. Sensing the air we breathe. http://www.opensense.ethz.ch/trac/, May 2012.

[KF11] Kannan Varadhan Kevin Fall. Network simulator 2. https://www.isi.edu/nsnam/ns/doc/index.html, November 2011.

[KJ10] Ujjwal Kumar and VK Jain. Arima forecasting of ambient air pollutants (o 3, no, no 2 and co). *Stochastic Environmental Research and Risk Assessment*, 24(5):751–760, 2010.

[KLM16] Xu Kang, Liang Liu, and Huadong Ma. Data correlation based crowdsensing enhancement for environment monitoring. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[Kou18] Ioannis Koukoutsidis. Estimating spatial averages of environmental parameters based on mobile crowdsensing. *ACM Transactions on Sensor Networks (TOSN)*, 14(1):2, 2018.

[KRZ+11] Sunyoung Kim, Christine Robson, Thomas Zimmerman, Jeffrey Pierce, and Eben M Haber. Creek watch: pairing usefulness and usability for successful citizen science. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2125–2134. ACM, 2011.

[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 *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 103–111. IEEE, 2014.

[KTS+17] Yasue Kishino, Koh Takeuchi, Yoshinari Shirai, Futoshi Naya, and Naonori Ueda. Datafying city: Detecting and accumulating spatio-temporal events by vehicle-mounted sensors. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 4098–4104. IEEE, 2017.

[KWW+16] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. Activeclean: interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment*, 9(12):948–959, 2016.

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

[KZX10] Matthew Keally, Gang Zhou, and Guoliang Xing. Watchdog: Confident event detection in heterogeneous sensor networks. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE*, pages 279–288. IEEE, 2010.

[LCC12] Mingqi Lv, Ling Chen, and Gencai Chen. Discovering personally semantic places from gps trajectories. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1552–1556. ACM, 2012.

[LGS15] Nabila Labraoui, Mourad Gueroui, and Larbi Sekhri. On-off attacks mitigation against trust systems in wireless sensor networks. In *IFIP International Conference on Computer Science and its Applications_x000D_*, pages 406–415. Springer, 2015.

[LHW07] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[LJS+17] Alina Lazar, Ling Jin, C Anna Spurlock, Kesheng Wu, and Alex Sim. Data quality challenges with missing values and mixed types in joint sequence analysis. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 2620–2627. IEEE, 2017.

[LMB10] Hyo-Sang Lim, Yang-Sae Moon, and Elisa Bertino. Provenance-based trustworthiness assessment in sensor networks. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks*, pages 2–7. ACM, 2010.

[LTTH16] Hong Liu, Ashwin Kumar Tk, Johnson P Thomas, and Xiaofei Hou. Cleaning framework for bigdata: An interactive approach for data cleaning. In *Big Data Computing Service and Applications (BigDataService), 2016 IEEE Second International Conference on*, pages 174–181. IEEE, 2016.

[LZW+17] Shengzhong Liu, Zhenzhe Zheng, Fan Wu, Shaojie Tang, and Guihai Chen. Context-aware data quality estimation in mobile crowdsensing. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[MB12] Anahit Martirosyan and Azzedine Boukerche. Preserving temporal relationships of events for wireless sensor actor networks. *IEEE Transactions on Computers*, 61(8):1203–1216, 2012.

[ME14] William Q Meeker and Luis A Escobar. *Statistical methods for reliability data*. John Wiley & Sons, 2014.

[MG13] Waze Mobile and Google. Waze. https://www.waze.com/, June 2013.

[MMH+15] Hayam Mousa, Sonia Ben Mokhtar, Omar Hasan, Osama Younes, Mohiy Hadhoud, and Lionel Brunie. Trust management and reputation systems in mobile participatory sensing applications: A survey. *Computer Networks*, 90:49–73, 2015.

[MW04] Stephen L. Morgan and Colleen G. Waring. Guidance on testing data reliability. http://www.auditorroles.org/files/toolkit/role2/Tool2aAustinCityAud$_{G}uidanceTestingReliability.pdf, January$2004.

[NP06] Mirco Nanni and Dino Pedreschi. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289, 2006.

[NRC+09] Kevin Ni, Nithya Ramanathan, Mohamed Nabil Hajj Chehade, Laura Balzano, Sheela Nair, Sadaf Zahedi, Eddie Kohler, Greg Pottie, Mark Hansen, and Mani Srivastava. Sensor network data fault types. *ACM Transactions on Sensor Networks (TOSN)*, 5(3):25, 2009.

[Org18] World Health Organization. Ambient air pollution - a major threat to health and climate. http://www.who.int/airpollution/ambient/en/, May 2018.

[PBKA08] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 863–868. ACM, 2008.

[PGWC16] Jaesung Park, Mikhail Gofman, Fan Wu, and Yong-Hoon Choi. Challenges of wireless sensor networks for internet of thing applications, 2016.

[PJP13] Sitthapon Pumpichet, Xinyu Jin, and Niki Pissinou. Sketch-based data recovery in sensor data streams. In *2013 19th IEEE International Conference on Networks (ICON)*, pages 1–6. IEEE, 2013.

[PK16] Amitangshu Pal and Krishna Kant. On the feasibility of distributed sampling rate adaptation in heterogeneous and collaborative wireless sensor networks. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2016.

[PK19] AMITANGSHU Pal and KRISHNA Kant. Smart sensing communication and control in perishable food supply chain. *2019, submitted to ACM TOSN*, 2019.

[PKK+11] Nikos Pelekis, Ioannis Kopanakis, Evangelos E Kotsifakos, Elias Frentzos, and Yannis Theodoridis. Clustering uncertain trajectories. *Knowledge and Information Systems*, 28(1):117–147, 2011.

[PKL12] Dong Woo Park, Junseok Kwon, and Kyoung Mu Lee. Robust visual tracking using autoregressive hidden markov model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1964–1971. IEEE, 2012.

[PP10] Sitthapon Pumpichet and Niki Pissinou. Virtual sensor for mobile sensor data cleaning. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5. IEEE, 2010.

[PPJP12] Sitthapon Pumpichet, Niki Pissinou, Xinyu Jin, and Deng Pan. Belief-based cleaning in trajectory sensor streams. In *2012 IEEE International Conference on Communications (ICC)*, pages 208–212. IEEE, 2012.

[PPK+12] Costas Panagiotakis, Nikos Pelekis, Ioannis Kopanakis, Emmanuel Ramasso, and Yannis Theodoridis. Segmentation and sampling of moving object trajectories based on representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1328–1343, 2012.

[PS07] Alfredo Petrosino and Antonino Staiano. A neuro-fuzzy approach for sensor network data cleaning. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 140–147. Springer, 2007.

[PSDG09] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.org/epfl/mobility/20090224, February 2009.

[PWC15] Dan Peng, Fan Wu, and Guihai Chen. Pay as how well you do: A quality based incentive mechanism for crowdsensing. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–186. ACM, 2015.

[RGB+17] Francesco Restuccia, Nirnay Ghosh, Shameek Bhattacharjee, Sajal K Das, and Tommaso Melodia. Quality of information in mobile crowdsensing: Survey and research challenges. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):34, 2017.

[RS11] S Reddy and V Samanta. Urban sensing: Garbage watch. *UCLA Center for Embedded Networked Sensing*, 2011.

[SBB13] Sukhwinder Sharma, Rakesh Kumar Bansal, and Savina Bansal. Issues and challenges in wireless sensor networks. In *2013 International Conference on Machine Intelligence and Research Advancement*, pages 58–62. IEEE, 2013.

[SBB+15] Stanislav Sobolevsky, Iva Bojic, Alexander Belyi, Izabela Sitko, Bartosz Hawelka, Juan Murillo Arias, and Carlo Ratti. Scaling of city attractiveness for foreign visitors through big data of human economical and social media activity. In *big data (BigData congress), 2015 IEEE international congress on*, pages 600–607. IEEE, 2015.

[SCHL13] Vijay Sivaraman, James Carrapetta, Ke Hu, and Blanca Gallego Luxan. Hazewatch: A participatory sensor system for monitoring air pollution in sydney. In *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*, pages 56–64. Citeseer, 2013.

[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.

[SHYL06] Yan Lindsay Sun, Zhu Han, Wei Yu, and KJ Ray Liu. Attacks on trust evaluation in distributed networks. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 1461–1466. IEEE, 2006.

[SKH+15] John Soldatos, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean-Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, et al. Openiot: Open source internet-of-things in the cloud. In *Interoperability and open-source solutions for the internet of things*, pages 13–25. Springer, 2015.

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

[SLMJ07] Bo Sheng, Qun Li, Weizhen Mao, and Wen Jin. Outlier detection in sensor networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 219–228. ACM, 2007.

[SPP+06] Sharmila Subramaniam, Themis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Online outlier detection in

sensor data using non-parametric models. In *Proceedings of the 32nd international conference on Very large data bases*, pages 187–198. VLDB Endowment, 2006.

[SRM+14] Victor Sucasas, Ayman Radwan, Hugo Marques, Jonathan Rodriguez, Seiamak Vahid, and Rahim Tafazolli. A cognitive approach for stable cooperative group formation in mobile environments. In *2014 IEEE International Conference on Communications (ICC)*, pages 3241–3245. IEEE, 2014.

[SSWX13] Bo Sun, Xuemei Shan, Kui Wu, and Yang Xiao. Anomaly detection based secure in-network aggregation for wireless sensor networks. *IEEE Systems Journal*, 7(1):13–25, 2013.

[STA16] STATISTICA. Number of smartphone users worldwide from 2014 to 2020 (in billions). https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, June 2016.

[TCA+14] Samia Tasnim, Mohammad Ataur Rahman Chowdhury, Kishwar Ahmed, Niki Pissinou, and S Sitharama Iyengar. Location aware code offloading on mobile cloud with qos constraint. In *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pages 74–79. IEEE, 2014.

[TCP+18] Samia Tasnim, Juan Caldas, Niki Pissinou, SS Iyengar, and Ziqian Ding. Semantic-aware clustering-based approach of trajectory data stream mining. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 88–92. IEEE, 2018.

[TMC13] Evangelos Theodoridis, Georgios Mylonas, and Ioannis Chatzigiannakis. Developing an iot smart city framework. In *IISA 2013*, pages 1–6. IEEE, 2013.

[TPI17] Samia Tasnim, Niki Pissinou, and SS Iyengar. A novel cleaning approach of environmental sensing data streams. In *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*, pages 632–633. IEEE, 2017.

[TPI+18] Samia Tasnim, Niki Pissinou, SS Iyengar, Abdur Shahid, et al. Reputation-aware data fusion and malicious participant detection in

mobile crowdsensing. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4820–4828. IEEE, 2018.

[TPI+19] Samia Tasnim, Niki Pissinou, S S Iyengar, Kianoosh G. Boroojeni, and Kishwar Ahmed. Rcod: Reputation-based context-aware real-time data fusion in mobile crowdsensing for iot. *Submitted to ACM Transaction on Sensor Networks (TOSN)*, 1(1):29, 2019.

[VCF+18a] Piergiorgio Vitello, Andrea Capponi, Claudio Fiandrino, Paolo Giaccone, Dzmitry Kliazovich, and Pascal Bouvry. High-precision design of pedestrian mobility for smart city simulators. In *IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018*, 2018.

[VCF+18b] Piergiorgio Vitello, Andrea Capponi, Claudio Fiandrino, Paolo Giaccone, Dzmitry Kliazovich, Ulrich Sorger, and Pascal Bouvry. Collaborative data delivery for smart city-oriented mobile crowdsensing systems. In *IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 2018*, 2018.

[VKG02] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684. IEEE, 2002.

[War13] Pete Warden. Data science toolkit. http://www.datasciencetoolkit.org/, May 2013.

[WLT10] Wei Wu, Hock Beng Lim, and Kian-Lee Tan. Query-driven data collection and data forwarding in intermittently connected mobile sensor networks. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks*, pages 20–25. ACM, 2010.

[YAS03] Yutaka Yanagisawa, Jun-ichi Akahani, and Tetsuji Satoh. Shape-based similarity query for trajectory of mobile objects. In *Mobile data management*, pages 63–77. Springer, 2003.

[YCL+14] Jia-Ching Ying, Huan-Sheng Chen, Kawuu W Lin, Eric Hsueh-Chan Lu, Vincent S Tseng, Huan-Wen Tsai, Kuang Hung Cheng, and Shun-Chieh Lin. Semantic trajectory-based high utility item recommendation system. *Expert Systems with Applications*, 41(10):4762–4776, 2014.

[YFW15] Yuan Yuan, Yuming Fang, and Lin Weisi. Moving object tracking with structure complexity coefficients. In *International Conference on Multimedia Modeling*, pages 59–70. Springer, 2015.

[YP12] M Yasodha and DRP Ponmuthuramalingam. A survey on temporal data clustering. *Int J Adv Res Comput Commun Eng*, 1(9):772–786, 2012.

[YS+09] Zhixian Yan, Stefano Spaccapietra, et al. Towards semantic trajectory data analysis: A conceptual and computational approach. In *VLDB PhD Workshop*, 2009.

[YSS18] Anes Yessembayev, Dilip Sarkar, and Faisal Sikder. Detection of good and bad sensor nodes in the presence of malicious attacks and its application to data aggregation. *IEEE Transactions on Signal and Information Processing over Networks*, 4(3):549–563, 2018.

[ZAS16] Yuxun Zhou, Reza Arghandeh, and Costas J Spanos. Online learning of contextual hidden markov models for temporal-spatial data analysis. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6335–6341. IEEE, 2016.

[ZBC+14] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[ZCWL07] Yongzhen Zhuang, Lei Chen, X Sean Wang, and Jie Lian. A weighted moving average-based approach for cleaning sensor data. In *27th International Conference on Distributed Computing Systems (ICDCS'07)*, pages 38–38. IEEE, 2007.

[ŽH13] Indrė Žliobaitė and Jaakko Hollmén. Fault tolerant regression for sensor data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 449–464. Springer, 2013.

[ZLH13] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444. ACM, 2013.

[ZSA+09] Anastasios Zafeiropoulos, Dimitrios-Emmanuel Spanos, Stamatios Arkoulis, Nikolaos Konstantinou, and Nikolas Mitrou. Data management in sensor networks using semantic web technologies, 2009.

[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 Z Sheng. An estimation maximization based approach for finding reliable sensors in environmental sensing. In *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on*, pages 190–197. IEEE, 2015.

[ZWL07] Wei Zhang, Ke-tai Wei, and Xi-mei Liu. Image denoising using multiple wavelet representations and local contextual hidden markov model. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 156–161. IEEE, 2007.

# VITA

## SAMIA TASNIM

| | |
|---|---|
| 2009 | B.Sc., Computer Science and Engineering<br>Bangladesh University of Engineering and Technlogy<br>Dhaka, Bangladesh |
| 2015 | M.S., Computer Science<br>Florida International University<br>Miami, Florida |
| 2019 | PhD Candidate<br>Florida International University<br>Miami, Florida |

PUBLICATIONS AND PRESENTATIONS

Tasnim, S. , Pissinou, N., Iyengar, S. S., Boroojeni, K. G., and Ahmed, K. " RCoD: Reputation-based Context-aware Real-time Data Fusion in Mobile Crowdsensing for IoT," *Submitted to ACM Transaction on Sensor Networks (TOSN)*, 2019.

Tasnim, S.*, Tang, Y. C.*, Pissinou, N., Iyengar, S.S., and Shahid, A., 2018, December. "Reputation-Aware Data Fusion and Malicious Participant Detection in Mobile Crowdsensing", *In 2018 IEEE International Conference on Big Data (Big Data)*, (pp. 4820-4828). Dec 2018 (*-equal contribution).

Tasnim, S. , Pissinou, N., and Iyengar, S. S. , "Real-time Identification of Movement Similarity ," *In Statewide Graduate Student Research Symposium*, Tallahassee, Florida, Apr 2018.

Tasnim, S., Caldas, J., Pissinou, N., Iyengar, S.S., and Ding, Z., 2018, March. "Semantic-aware clustering-based approach of trajectory data stream mining," *In 2018 IEEE International Conference on Computing, Networking and Communications (ICNC)*, pp. 88-92. Mar 2018.

Tasnim, S. , Pissinou, N., and Iyengar, S. S. , "Real-time Identification of Movement Similarity - a Semantic-aware Approach," *In Grace Hopper Celebration of Women in Computing*, Houston, Texas, Sept 2018.

Tasnim, S., Pissinou, N., and Iyengar, S.S., (2017). "A Novel Cleaning Approach of Environmental Sensing Data Streams," *The 14th IEEE Annual Consumer Communications Networking Conference (CCNC, Poster)*, pp. 632-633. Jan. 2017.

Tasnim, S. , Ahmed, K., Pissinou, N., and Iyengar, S. S. , "Offloading Code Efficiently on Mobile Cloud," *In Grace Hopper Celebration of Women in Computing*, Houston, Texas, Oct 2016.

Tasnim, S., Chowdhury, M.A.R., Ahmed, K., Pissinou, N. and Iyengar, S.S., "Location aware code offloading on mobile cloud with QoS constraint", *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pp. 74-79. Jan. 2014.

Begum, N., Tasnim, S., and Naznin M., "Optimal queries processing in a heterogeneous sensor network using multicommodity network flow method", *In 2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management*, (pp. 665-670).