Community-Oriented Architecture for Smart Cities

By

Roozbeh Jalali

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Computer Science

Faculty of Science

University of Ontario Institute of Technology

December 2017

Abstract

With the widespread use of smartphone devices, a surge in mobile sensing, progress in wireless communication and networking techniques, as well as the development of the Internet of Things (IoT) and cloud computing, mobile-based community sensing has turned into a leading paradigm for pervasive sensing. Smartphones with embedded sensors have become ubiquitous devices carried by millions of people. Community sensing empowers individuals to collectively sense, analyze and share local observations and mine data in order to determine and map phenomena relating to real world conditions by using mobile devices across many applications, including transportation and healthcare. While there are currently many tools and frameworks that allow researchers and developers to collect and analyze data at the individual user level, a parallel framework for data collection and analysis at the community level does not yet exist. Such a framework would provide the functionality to create various models for building smart city applications for urban planning, sustainable communities, transportation, public health, and public security.

This thesis presents a review of current smart city network architectures, along with their associated technologies, and proposes an architecture for the smart city and its services while considering communities as the main part of the design. Of the different components of the proposed architecture, two are vital for enabling a community structure for the smart city. These two components are community detection and data aggregation. This thesis proposes new methods for community detection and analysis using graphs and clustering algorithms based on the sensor data collected from individuals' smartphones and IoT sensors. As far as can be ascertained, the proposed method is the first to transform the time series data collected from individuals' smartphones to correlation networks for community detection. The proposed methods leverage not only the individuals’ groups but effectively discover communities of common interest. Two different case studies were conducted in this thesis in order to show the performance of the proposed methods. In these case studies, the data collected from individuals’ smartphones and vehicles are used and communities of individuals, based on their movement patterns and similarities, are detected. The performance evaluation shows that the proposed methods effectively identify the individuals’ communities with good accuracy.

Dedicated to my mother, father, brother and

my wife, Shohreh,

without whose support and encouragement none of this would have been possible.

Keywords

Community-Oriented Architecture, Big Data Analysis, Community-Aware Sensing, Internet of Things, Smart Cities, Community Detection, Clustering.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Dr. Khalil El-Khatib and Dr. Carolyn McGregor, for their supervision, guidance, and continued support throughout my research. I would like to thank them for their motivation, immense knowledge and the insightful discussions that have helped me to accomplish this thesis.

I would like to extend my appreciation to my committee members, Dr. Shahram Heydari and Dr. Richard Pazzi, for evaluating my thesis and providing their invaluable comments and feedback.

I would also like to extend my thanks to Dr. Daniel Hoornweg and my friends, Dr. Alireza Izaddoost and Mehran Kamkarhaghighi, for their encouragement and support throughout my research.

I also wish to express my appreciation to the University staff for their assistance, generous help and administrative support. In particular, I must make special mention of Mrs. Catherine Lee, for editing and proofreading of this thesis.

Finally, I would like to express my greatest gratitude to my family: my dear wife, Shohreh, my mother, my father and my brother for their unconditional love and support during my study.

Table of Contents

List of Figures

List of Tables

Abbreviations

IoT	Internet of Things
ICT	Information and Communication Technology
MIT	Massachusetts Institute of Technology
RFID	Radio Frequency Identifier
EPC	Electronic Product Code
ITU	International Telecommunication Union
WSN	Wireless Sensor Network
NFC	Near Field Communication
MCS	Mobile Crowd Sensing
SOA	Service Oriented Architecture
ICU	Intensive Care Units
ECG	Electrocardiogram
EMG	Electromyography
WHMS	Wearable Health Monitoring System
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
CSM	Community Sensing and Mining
MCCS	Mobile-based Community and Crowd Sensing
SCN	Sensor Correlation Network
UCN	User Correlation Network
DTW	Dynamic Time Wrapping
NGA	Newman’s Greedy Algorithm
GTA	Greater Toronto Area
RI	Rand Index
JI	Jaccard Index

NMI	Normalized Mutual Index
MI	Mutual Information
CDR	Call Detail Records

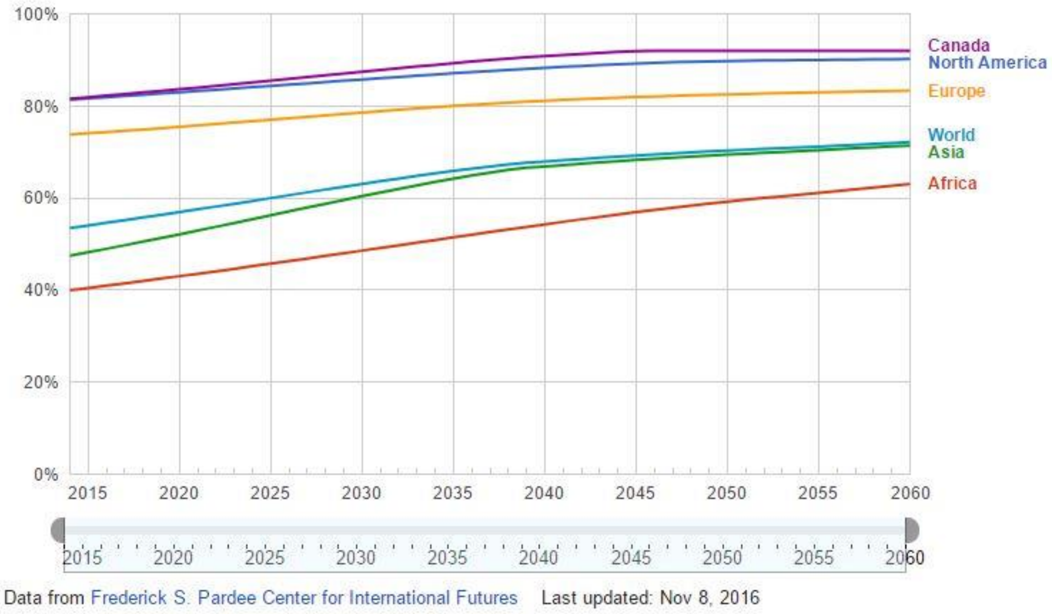
Chapter 1

1 Introduction

1.1 Overview

Today, more than half of the world's population spend their lives in cities (Figure 1), a number which is expected to reach 85 percent by 2100 [1]. If considering only North America, “more than 82 percent of the population already lives in urban areas” [1]. Increasing population density in urban environments demands adequate provision of services and infrastructure [2]. This increase in urban populations will present major challenges, including increased air pollution, traffic congestion, housing requirements, health concerns, energy and waste management. Failure to plan and manage these projected challenges could lead to increased urban crime rates, slums, and deteriorating quality of life in the urban environment. Opportunities abound for these challenges to be addressed through the integration of various Information and Communication Technologies (ICT) into the fabric of the urban environment.

The smart city, as a new paradigm in the ICT domain, provides the infrastructure for citizens to easily access many services, as well as for governing bodies to intelligently manage and control the resources in a city [2]. Smart cities use ICT to sense, analyze and integrate the information necessary for city administration[3]. As the population of cities grows [1] and the boundaries expand, the concept of the smart city is gaining momentum on the agenda of governments around the world, and can be seen as a key mechanism for transforming traditional cities into becoming more efficient and viable [2, 4].



Data from Frederick S. Pardee Center for International Futures Last updated: Nov 8, 2016

Figure 1. Population, urban percent [5]

Two major developments that occurred in ICT have impacted how cities are managed: the Internet of Things (IoT) and mobile telecommunication. The term IoT was introduced in 1999 by Kevin Ashton while he was working at the Massachusetts Institute of Technology (MIT) Auto-ID lab [6]. The concept of the IoT was then expanded by the MIT Auto-ID Centre and linked to the Radio Frequency Identifier (RFID) and Electronic Product Code (EPC) in 2001 [7]. The IoT was later expanded by the International Telecommunication Union (ITU) in 2005 [8]. The premise behind this paradigm is the ubiquity of different kinds of objects or things such as tags, RFIDs, sensors and mobile phones [9] and the main goal of the IoT is to connect these objects that can act independently and intelligently connect with each other with minimum human intervention [9]. It is predicted that the IoT will penetrate ordinary life by 2025 [9]. As Figure 2 shows, the IoT is situated in the peak of Gartner's 2015 hype cycle for emerging technologies [10].

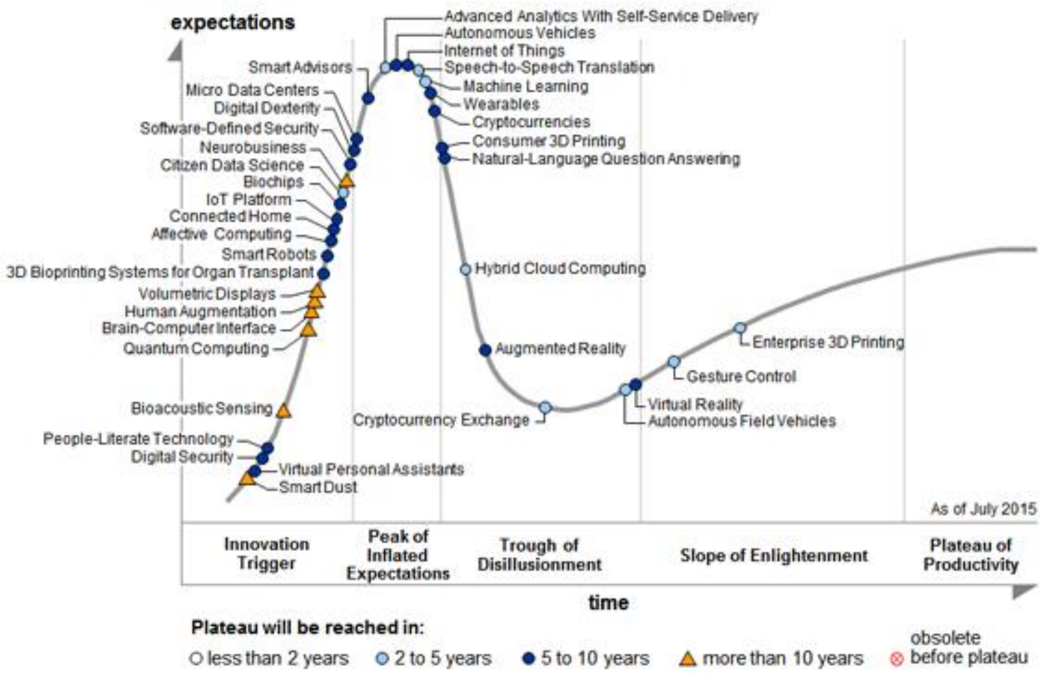


Figure 2.Cycle for emerging technologies, 2015 [10]

In parallel to the developments in the IoT, mobile communication has expanded on a global level over the last few years, so that today there are more than 6.8 billion mobile phones in use worldwide [11], including 2 billion smartphones. Mobile devices, particularly smartphones and tablets, have various built-in sensors, with many applications that can use and share the data collected from these sensors. Each smartphone contributes to the vast amount of sensor data generated every day.

With the rapid growth of the IoT, mobile communication and cloud computing [12], there is high potential to develop real-time applications to monitor and mine data collected from large environments. There exist many challenges, such as how to collect and analyze the large amounts of data from various sources of information with different types of technologies, and how to deliver that data to the different applications, while considering the requirements of each application [2]. A more important challenge is how to use ICT to understand how users consume resources in an urban setting as well as how to use ICT to help users, individually and collectively as communities, to manage resource usage. Using a community approach, where everyone has sensing and computing capability, can result in collectively sharing and extracting data to measure common phenomena. This integration of ICT into the fabric of the city forms what is commonly referred to as the smart city.

While there are currently many tools and frameworks that allow researchers and developers to collect and analyze data at the individual user level [13-19], a parallel framework for data collection and analysis at the community level does not yet exist. Such a framework would provide the functionality for creating various models to build smart city applications for urban planning, sustainable communities, transportation, public health, and public security. The major open question is how to sense beyond the single individual and leverage individual data to build knowledge at the community level.

The focus of this thesis is to design a reference architecture for the smart city and its services while considering communities as the main part of the design. This architecture and the associated framework is scalable, interoperable, secure and homogeneous from both the service provider and service consumer perspectives. Within this research, the framework will be instantiated in a platform enabling community-based analytics. The platform will be demonstrated within the context of similarity over trajectory case studies.

1.2 Problem Statement

The authors in [20] claimed that “Urban areas can provide many advantages, such as better education, health services, entertainment, political participation and freedom from traditional norms”. Such advantages of urban living cause continuous migration from rural to urban environments, leading to larger cities. Growing populations in urban areas demand more resources and better services [21]. The challenge is how to optimize the resources and increase the quality of services while city populations are growing. Cities are becoming increasingly complex as large networks of sensors are deployed across urban landscapes [22]. Moreover, as cities have begun to integrate technology into their systems, it has become apparent that sensor technology in particular has the potential to optimize, and even enhance, the services that cities could provide to their residents [23].

Smart cities need to deal with many issues, ranging from traffic and waste management to healthcare and security. To address all these problems, it is very important for smart city developers and researchers to plan how to collect data from various sources of information and distribute them among service providers. It is also necessary to plan how to deliver the appropriate services with minimum cost to consumers. In addition to the above factors, there is a need to have analyzing power to mine data and provide more services.

In the next decade, information and communication technology will pervade daily life. Smartphones can play a significant role in this scenario because they have a variety of sensing, computing and communication capacities as well as the capability to bridge to other objects and devices. The emergence of the smartphone as a convenient sensing platform has shifted the level of smart city sensing research. With the variety of sensors possessed by each smartphone, it has become possible to easily build a sensing platform in the smart city, hence coping with most of the sensing barriers to building a large-scale sensing system.

The focus on smartphone sensing design is still on an individual level but collecting and analyzing individual data ignores an important dimension of personal life, namely: how others, such as friends and family, influence an individual’s life. In fact, individuals are affected by the behaviour of others and are tightly connected via various communities. Therefore, in collecting and analyzing community data, one can have a better understanding of complex community dynamics as well as be able to more efficiently manage city resources. Moreover, sensing large scale phenomena is sometimes beyond the scope of individuals and needs to be collectively measured by a large group of individuals. Finally, the aggregation of data from a group or community usually results in better decisions than those made by individuals. It is important to also realize that citizens can be engaged in different communities at different periods of their lives as well as concurrently at any given time. Such communities can be formed according to people's interests, locations, and common goals. These communities can be varied, from healthcare to transportation to even same brand car owners. Therefore, the focus of this research is to use ICT to build communities and to share information among community members. The goal is not just to exchange the information among the parties but to involve the individual’s environment and belongings. While current smart city research relies on individual sensing and analyzing, the smart city can benefit from community and crowd sensing in various aspects, including diversity of opinion and opinion aggregation.

Strong urban management and city development relies on adequate monitoring of community dynamics for decision making. To reach this goal, one approach is to leverage distributed sensors, such as sensor networks, to obtain real world conditions [24]. However, sensor networks and other traditional techniques have never successfully achieved this goal because of limitations such as high installation and maintenance costs, inappropriate coverage and fixed sensors [25]. Mobile-based community sensing uses the existing sensing and communication infrastructure of smartphones, hence eliminating the cost of deployment. The inherent mobility of smartphones and mobile devices also provides better sensing coverage than traditional fixed sensor networks.

By using ICT, there is an opportunity to develop a community-aware smart city using mobile phone platforms on any scale, but the major challenge is how to design a scalable platform which would support the hundreds of thousands of user demands and integrate these demands with modern technology in order to provide adequate services.

This research comprises the deeper integration and coupling of communities of people and the sensing systems that serve them in a way that will improve the quality of life for citizens and optimize resource consumption.

1.3 Research Challenges

Using ICT to build communities in smart cities holds several challenges, including but not limited to the following:

- **Architecture**: Based on advancements in sensor and wireless networks, there is potential to connect objects to each other. Each object can generate a large amount of information and share this information with other objects. Together, these objects create the IoT. The base of most IoT applications and services is dissemination and sharing of information among the respective parties. The challenge is that most of the current IoT architectures were designed to handle host-centric communication and do not effectively support data-centric communication [26]. The current Internet-similar architecture is not efficient in terms of network resources and can also increase communication overheads [23, 26] .

- **Heterogeneity**: Different IoT devices will be created by different companies and manufacturers. Each device can have distinct characteristics and different functionalities. Having all the devices seamlessly work with each other is a major challenge.
- **Varied Communities and Human Groupings**: As people interact with each other and collaborate in social activities in communities, grouping individuals and simplifying the interactions among them can be a challenge [24]. Designing the mechanism for community formation along with an automated method to identify the user communities is another challenge in smart cities. Furthermore, while the emergence of smartphones that are equipped with multiple sensors has shifted the sensing research, there are numerous challenges regarding both how to analyze the sensor data and how to detect the communities of individuals from their respective sensor data. Designing algorithms that can identify the similarity of people according to the similarity of their sensor data is an additional challenge.
- **Data Dissemination**: A smart city will include diverse communities, from healthcare to transportation, therefore a generalized framework to leverage the communities within the sensing system is required. Common methods of data collection and dissemination need to be developed in a way that facilitates community sensing.
- **Data Quality and Redundancy**: In community sensing, most of the tasks are performed by sensing the environment via mobile devices. Therefore, based on the

dynamic conditions of mobile devices, the accuracy of sensing data is different from one device to another. The quality of sensed data constantly changes due to mobility, energy levels and communication channels. Moreover, because of human involvement, individual preference can affect sensing and data quality. Equally important, more than one participant can be involved in the sensing process which can cause data redundancy. Redundancy can also lead to inconsistency due to the differences in sensing capabilities.

- **Incentive Mechanisms**: In community sensing, because humans are involved and sensors are usually possessed by different individuals, participation in the sensing process depends on individual incentive, carries a cost and incurs energy. Without strong incentive mechanisms, participants may not be involved in the sensing process. Therefore, finding a way to motivate individuals to be active in the sensing process is an additional challenge. Furthermore, if money is part of the incentive mechanisms, participants are more likely to deceive the system to gain more benefit from it.
- **Predictive Analytics**: One of the major benefits of the smart city is providing a platform to collect information from citizens in different ways. A smart city can benefit by integrating the collected information. Such information can be varied from traffic and health data, as well as environmental conditions, to the movement patterns of citizens. As the volume of this data is expected to steadily increase, analytical tools are required to combine knowledge from different and related sources to draw a picture of a live city in motion. Analyzing the collected data makes it possible to predict the future. Prediction data can be used to build models of the various city systems and their interactions, which can be used for urban planning to improve both the quality of life for citizens as well as the economic growth of the community. Developing and

designing these analytical tools, which can precisely combine information and predict the future, is one of the challenges of any smart city.

- **Big Data and Scalability**: As the number of sensors in a city grows to the hundreds of thousands, these sensors will generate massive datasets. If analyzed in real-time, this data can be used to model movements, actions, and needs in a way that enables cities to appropriately respond. This process of collecting and analyzing massive datasets is commonly applied in Big Data systems [27]. Big Data makes use of powerful modern processing capabilities to analyze large datasets which would be impossible to analyze through use of efficient search algorithms. While many smart city systems that currently operate do not require the implementation of Big Data schemes to interpret their results, within the next few years the number of data streams will grow; as such, the need for Big Data systems is inevitable [11].
  - **Cost of the Sensing Infrastructure**: One of the major limiting factors for smart cities is the cost of deploying sensors in a city. As an example, the sensors that Chicago currently deploys to detect gunshots cost roughly $100,000 per 1.5 miles of coverage [27]. Considering that these sensors only serve a single application, it is difficult to imagine them being deployed across a large metropolitan city. While cost is not the only issue, it is directly related to many subsequent issues. Because the technologies used inside sensory technology is subject to Moore's law, as new technologies emerge with improved processing capabilities and additional functionality, older sensors may quickly become obsolete and will require replacement.
  - **Privacy**: The provision of privacy is one of the major challenges in developing the smart city. As some sensors collect personally identifiable information such as

location, time, and health information, providing privacy is very important for the smart city. The issue of ubiquitous sensors has raised significant concerns among privacy advocates as to how and by whom this large amount of data is used, how long it will be kept, and who can access it [9].

1.4 Research Objectives

The main goal of this research is to design a smart city platform which will support community services. One of the most important tasks to enable community services is to identify communities of individuals and communities of common interest. Since a smart city has the proper infrastructure to collect data from individuals and their belongings, identifying the individuals' communities can be performed by analyzing their data, which is collected from their sensors. However, individually analyzing users' data cannot reveal the communities and will only give some information about the corresponding user. Therefore, there is a need to design a proper framework and platform to aggregate data from different users and collectively analyze them. Furthermore, the designed platform should support different types of communities such as social, infrastructure and environmental. Moreover, IoT can help in the smart city to more easily collect individual data. Therefore, the designed platform should support IoT applications and their requirements. The objectives to achieve this goal are outlined as follows:

- **Developing New Community-Oriented Architecture and Framework for Smart Cities**

The primary objective of this research is to develop a new architecture for the smart city, from the sensing layer to the application layer, by considering communities as the main part of the design. This architecture enables city management, community service providers and citizens to have access to real time and historic data that has been gathered using various sensory mechanisms, in order to analyze and make decisions for future planning.

- **Develop a Community Sub-layer**

The second objective of this research is to develop a sub-layer which can enable virtual communities. This sub-layer extracts the context and forms the communities. In addition, it can automatically identify the communities of individuals and communities of common interest based on users' various activities or allow individuals to self identify within a given community. Such a framework can benefit from various models in order to build smart city applications for urban planning, sustainable communities, transportation, public health, public security, and commerce.

- **Aggregate Data Analysis**

The individual data analysis only analyzes the data for a single person or a single device. Community-based analytics can rely on analyzing data from a group of people or a collection of devices. Community-oriented analytics can aggregate the data from each individual and identify useful patterns. Therefore, the third objective of this research is to perform multi-modal analysis of large datasets from a smart community and to visualize and analyze this massive heterogeneous aggregate data.

1.5 Research Contribution

This thesis proposes a new multi-layer architecture for smart cities by considering communities as the main part of the design. This architecture enables city management, community service providers and citizens to have access to real time and historic data, which has been gathered using various sensory mechanisms, in order to analyze and make decisions for future planning. The architecture includes a layer which can enable community analysis. This layer extracts the context and forms communities. In addition, it can automatically identify the communities based on users' similar activities and common interests. Such a framework can benefit from various models to build smart city applications for urban planning, sustainable communities, transportation, public health, public security, and commerce. This architecture and its associate framework has tremendous potential to solve some of the challenges previously mentioned.

To sum up, the contribution in this thesis is as follows:

- A comprehensive study of IoT and its enabling technologies, the smart city, the context-aware system and community sensing and mining is provided in Chapter 2.
- A community-oriented architecture and framework have been proposed for the smart city in Chapter 3.
- New methods for community detection and analysis using graphs have been proposed, developed and evaluated in Chapter 4.
- An additional method of community detection beyond the graph, that is by conducting a case study on detecting communities of common interest that demonstrates an infrastructure based community, is provided in Chapter 5.

1.6 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 presents a literature review on IoT, the smart city, community sensing and various smart city enabling technologies. The proposed network architecture for a smart city and its associated framework is presented and discussed in Chapter 3. Chapter 4 includes an explanation of the communities and discusses the benefits of community detection in a network. The proposed method for community detection using graphs is also presented along with a real case study and experimental results. This chapter demonstrates how social communities can be framed in the designed framework. Chapter 5 presents the proposed methods for community detection using clustering algorithm Chapter 5 also includes a case study outlining how historical GPS data on privately-owned vehicles in Changsha, China, were collected and used in an algorithm developed to match riders with close temporal and spatial origin and destinations. This chapter demonstrates how different methods other than graph analysis can be used to detect communities. Chapter 6 concludes the work by summarizing the main findings and suggesting future areas of exploration.

Chapter 2

2 Literature Review

This chapter presents an extensive study of the previous work in two important topic areas: (1) the IoT and smart cities; and (2) community sensing and mining. The focus of the first part of the chapter is more on the IoT and smart city architectures, middleware and enabling technologies. Consequently, existing IoT architectures, along with middleware technologies and content dissemination techniques, are reviewed. Because this thesis is focused on information centric architecture, context-aware computing for the IoT and sensor networks is also reviewed. As the aim of the research of this present work is to provide functionality to support community structure, previous studies on community mining and sensing in the smart city are also reviewed.

To identify resources for the literature review, the IEEE, ACM, Science Direct, ProQuest and Google Scholar databases were used. Approximately 1000 related papers (journal and conference papers) were found by searching topic-related keywords such as: smart city; Internet of Things; IoT architecture and middleware; Big Data analysis; community sensing; participatory sensing; remote healthcare platform; and wearable technology. By just reviewing the titles, the list was narrowed down to 120 papers. Furthermore, the abstract of each paper was reviewed and the 70 most relevant papers were chosen. By following the citations in each paper, the list was expanded to approximately 90 papers.

2.1 Background on the Internet of Things and Smart City

In recent years, there has been extensive research in the domain of the IoT. These research studies have tried to make the IoT feasible [28] but the success depends on emerging IoT enabling technologies, expanding IoT visions and developing new applications for solving city management problems. A brief overview of the above-mentioned conditions is provided in the following subsections.

2.1.1 IoT Vision

The name "Internet of Things" is composed of two principal words, *Internet* and *things*, which introduce the concepts of Internet-oriented vision and things-oriented vision [9]. The simple definition of the IoT comes from these two perspectives which consider *things* as very simple items, such as RFID tags, which are able to connect and communicate with each other [9]. Internet-oriented vision is more related to networking and communications, and with how things and objects can use different network protocols to communicate with each other, while thing-oriented vision is more focused on objects and how these objects can be combined into a common framework [9, 29]. In the network-oriented vision of the IoT, many alliances and councils try to adapt the IoT to IP technology. As an example, 6LoWPAN and Internet0 have followed the approach of reducing the complexity of the IP stack to achieve a protocol designed to route IP over everything. IoT definitions can differ according to the aspect selected by researchers, industries, and business alliances. The combination of the Internet and things reveals a third vision known as semantic vision. IoT semantically means a global network of objects connected together. The number of objects in the future Internet will increase dramatically, therefore organizing this number of things is one of the challenging issues in IoT. Semantic vision plays an important role in solving this challenge [9].

In this thesis, the definition that explores the IoT vision from both perspectives (Internet and things) as defined in [30] is followed, namely: “The IoT allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service”. This definition effectively defines the all-encompassing vision of the IoT.

2.1.2 IoT Enabling Technologies

Enabling technologies have a crucial effect on actualizing the IoT into the real world. In this section, two important enabling technologies are reviewed in order to provide a comprehensive survey of each technology. These two technologies have significant influence in the accelerating adoption of the IoT.

2.1.2.1 Sensing and Identification

Wireless technologies play a crucial role in the sensing and identification of objects. There is potential to install a wireless adapter on all objects around us and enable the IoT concept. Sensor networks will also play a vital role in the IoT and will work side by side with RFID to better track the status of things such as location, temperature and movement. [9].

- **Wireless Sensor Network**

The Wireless Sensor Network (WSN) is one of the most crucial technologies for the realization of the IoT. The authors in [31] stated that “WSNs are made of distributed autonomous sensors that control and check physical and environmental conditions” [31].

These sensors are equipped with wireless transmitters to communicate with each other [31]. Advances in sensor technology and microprocessor design help to create smaller and more durable sensors for WSNs, which can accelerate the IoT actualization.

- **RFID**

RFID tags are one of the important elements of the IoT. RFIDs are used by many things such as merchandise in stores, maps, and posters. Even passports have RFID tags to prevent forgery and to identify false documents. Since mobile phones are increasingly being equipped with Near Field Communication (NFC) technology, they may be adapted to read RFID tags and extract from them useful information such as location and price [6].

- **Collaborative Sensing**

Collaborative or participatory sensing is a new method for sharing sensory information among individuals. Building on the recent growth in smartphones and mobile devices, people around the world can easily sense their environment and share the information via other persons or machines [24]. Collaborative sensing uses the cloud to aggregate and fuse data for crowd intelligence extraction [32].

2.1.2.2 Advances in connectivity and network

The IoT consists of a very large number of nodes, which could have been a problem in IP availability if IPV4 had been used. The IPv6, which widely accepted the 128-bit Internet scheme to provide the needs of the IoT, was created to solve this problem. Once the objects are connected, all IoT devices need a network to communicate with other devices. Various types of communication standards and network technologies are available, each with their own strengths for certain applications. Some, such as Bluetooth and ZigBee, are useful for short-range communication while others, such as LTE and Wi-Fi, are useful for long-term communication. All these technologies together bring the IoT concept into reality [29].

2.1.3 Smart City Applications

The smart city provides good infrastructure for developing many applications. These applications can improve the quality of life of citizens and simplify the services that are offered by cities. Smart city infrastructure as provided by the IoT has the potential capacity for running many applications, including transportation, healthcare, the smart environment, urban sensing, social recommendations and public safety. Smart city development can be divided into three sections: infrastructure, platform development, and application development. Application development is one of the important factors for smart city development [33] while the development of Mobile Crowd Sensing (MCS) has resulted in various innovative applications. In these applications, actuators and sensors, set up on a vast scale throughout the network, will allow the collecting of real-time environmental information. Table 1 lists the main smart city and MCS application categories, with an example from each category.

Table 1. Smart city application categories

Application Category	Subcategory (example)	Related Work
Transportation	Traffic monitoring, public transportation, road conditions, mobile ticketing	Conway-Beaulieu and Jalali [22]
Healthcare	Public health, wellness, activity detection	DietSense [34]
Public safety	Crime analysis and prediction	Crime Prediction [35]
Smart urban sensing	People behaviour/social monitoring	UBhave [36]
Recommendation	Trip advisor, activity recommendation, place recommendation	Zheng and Xie [37]
Environment monitoring	Air pollution, noise monitoring	CommonSense [38]

The following subsection provides a review of the top three useful application categories which can run on top of the proposed smart city architecture.

- **Transportation**

In recent years, many vehicles, such as cars, trains, and bicycles, have become equipped with sensors. These types of sensors can collect information about the location, speed, and status of vehicles. Such information allows for traffic management, route creation and decreased air pollution in cities. As an example, many applications exist in the transportation category, including assisted driving, mobile ticketing, environment monitoring, individual travel planning, road conditions and augmented maps. As many applications for single person purposes already exist, there is potential ability to develop applications for community use. For example, by collecting an individual's location in the neighborhood community and aggregating this data, public transportation services can be managed and improved [22].

- **Public Safety**

Cities with larger populations often have higher crime rates. However, large cities that also suffer from poor education and high unemployment rates can create a breeding ground for criminal activity [39]. One of the traditional ways to prevent and report crime activity is the neighborhood watch program. This method needs a community association, involving a group of residents who are assigned to monitor and report criminal activity to the authorities. Due to issues of unavailability, distraction and limited perception, this method is often inconsistent and of limited effectiveness [40]. In recent years, crime recording and reporting has been carried out using technological methods which have enhanced efficiency of output. As well as creating a record of crime details, this data also provides any existing relations between crime scenes and an offender's modus operandi [41]. Criminal analysis involves a very careful evaluation of the location, time, and type of crime that has been committed at a building or neighborhood, or within a city or country. Crime statistics, risks and probabilities are very much the essence of criminal analysis. As an example, by employing appropriate data collection, pre-processing and query techniques, and finding patterns and trends in the language of tweets, it would be possible to reveal the relationship between tweets and crimes in given locations to gain knowledge of a particular city's crime rate [35].

- **Healthcare**

In non-clinical environments, global developments in smart healthcare and health monitoring are progressing at a rapid level. Wearable technology is one example of the promising technologies that can help us conduct remote healthcare monitoring in a smarter way. As a result of the current ability to decrease sensor size and design accurate sensors, wearable sensors are becoming a growing trend [42].The data gathered from sensors is important as it can then be processed into some form of useful information. Data mining is one of the viable methods applicable to the processing of the significant volume of health data, such as vital signs, that can be collected from wearable sensor networks. The issue is whether working on integrated health data and mining is of benefit to the community, and whether it is actually relevant [42]. At-home sensor monitoring systems and wearable ubiquitous technology form an instrumental component of a smarter city. Patients are becoming more active in taking care of their daily lives and improving their health conditions. The importance of human health sensing technology has been speculated to address stress management, preventive attention to falls, chronic disease supervision, and tele-monitoring rudimentary physiology in rural locations. Using intelligent sensor-based technology to improve healthy living is a key aspect of the proposed research plan.

2.1.4 Architectures and Middleware for the Smart City and the IoT

To design an architecture for the IoT and the smart environment, the characteristics and limitations of objects in the IoT, such as sensing, energy, connectivity and computation, should be carefully considered. Sensor networks are an important aspect of the IoT and play a crucial role in IoT architectures. Therefore, sensor network architecture affects the IoT architecture, which consequently affects smart city architecture. Figure 3 shows the relationship between sensor networks and the IoT [43].

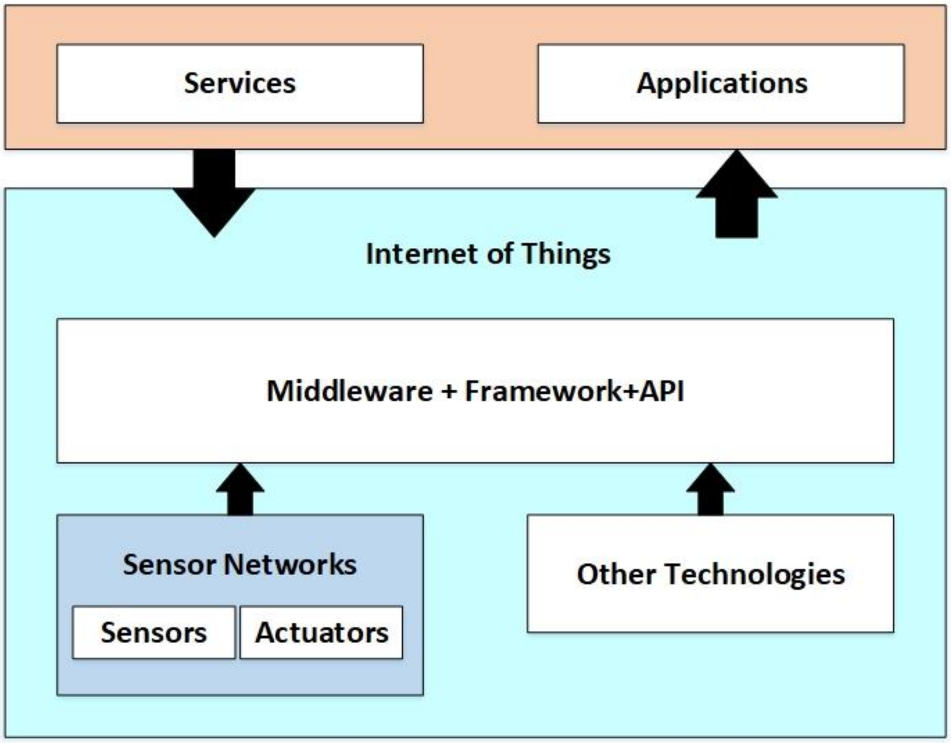


Figure 3. Relationship between IoT and sensor networks

Sensor networks follow three main architectures: flat architecture, two-layer architecture and three-layer architecture [43]. In a flat architecture, sensors are connected to each other by using multi-hop fashion and data transfers from the sensors to the sink node. In the two-layer architecture, more than one mobile sink node exists to gather data and transfer it to the upper layer. Finally, the three layer architecture uses the Internet or any other wide area network to connect multiple sensor networks to each other [43]. The IoT follows a three-layer architecture in the sensing layer to overcome most of its challenges, such as scalability and heterogeneity.

Many research studies focus on designing the architecture, middleware, and framework for smart cities and the IoT, including IoT-A [44], COMPOSE [19], FIRE [45], FIND [14], and BUTLER [13]. These studies deal with large interconnected objects and heterogeneous networks. Some adopt layered architecture while others adopt Service Oriented Architecture (SOA). While each of these projects tried to solve the challenges related to IoT, including interoperability, heterogeneity, security, scalability and platform portability, none of them tackled all of the challenges. IoT-A [44] architecture mainly concentrates on building an architecture model, along with addressing security, management and protocol-level communication issues of the different components of the architecture. COMPOSE [19] tries to enable new services and combine the virtual and physical world by converging the internet of services with the IoT. Inside the COMPOSE architecture, smart objects are associated with services that can be merged, controlled, and federated in a standardised way to simply and rapidly create novel applications [19]. “BUTLER is a European project that aims at enabling the development of secure and smart life assistant applications” [13].

Most of the research studies on the smart city focus on developing an architecture for a specific purpose or application [46-51], from healthcare to traffic management. As an example from healthcare, in [49] the authors proposed an architecture called KNOWME, which is a 3-tier platform designed to monitor single user health status. The problem of KNOWME is scalability; because most analytical jobs are performed on mobile phones, based on mobile phone constraints such as computation power and memory, the architecture is not scalable. In [50], health signals are collected via body sensors and transmitted to caregivers via the Internet or a mobile gateway but the issue is that the design framework is specified to user requirements. Artemis Cloud [51] is another Big Data platform for online health analytics. However, Artemis does not analyze community data but merely supports personalized online health analytics in a clinical environment.

Many researchers have proposed a layered based architecture for the IoT and smart cities. For example, in [52] and [53], the authors suggested an architecture similar to OSI architecture. In [54], the author suggested a layered based architecture but this failed to support the community model.

DIAT [18] is an architecture for the IoT which solves scalability, heterogeneity, interoperability and security. DIAT, which uses a distributed model to make a system scalable, also uses a virtual object layer to tackle the heterogeneity problem. The main purpose of DIAT is to use the layering model for its architecture as an IoT daemon (Figure 4). DIAT also uses the cross-layer security model to ensure security and privacy. Similar to other architectures, DIAT suffers from a lack of community support.

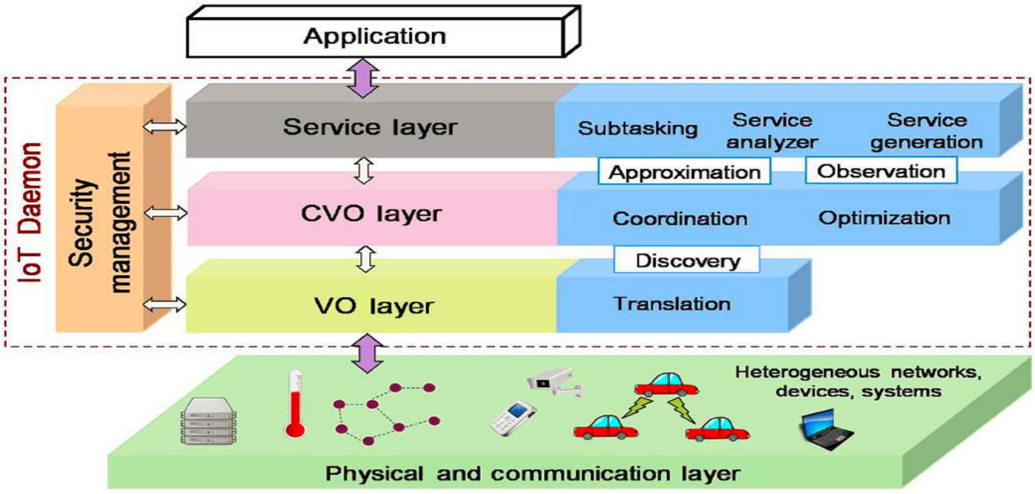


Figure 4. DIAT architecture [18]

Some research, such as the works in [55] and [56], focuses on the development of an architecture for the smart city but fails to support scalability and interoperability.

From the IoT perspective, all entities and objects can be seen as service providers [18]. In reality, some of the objects cannot provide a complete service for the consumer, therefore need to integrate their services and compose new services. The SOA based approach is an appropriate solution for such cases because objects and service providers can exchange information with each other in a simple way and without human intervention. Hence, many research studies [57-59] agree that service oriented architecture is an approach that is applicable for smart cities and the IoT.

Recently, the SOA approach has been applied to centralized IoT middleware [58]. In SOA based middleware, where services are presented as a web service, applications can perform complex tasks by composing services provided by different objects or devices.

Generally speaking, a middleware is a software or set of sub-layers located between the application layer and the object layer (Figure 5) and which helps to reduce the complexity of the heterogeneous nature of the objects. In addition, it simplifies the programmer's job of knowing the exact characteristics of various underlying objects for writing appropriate applications. Because of the variety in protocols and data format provided by different devices, middleware is used to provide a universal interface to application developers in order to facilitate the development.

Middleware in the IoT connects heterogeneous application domains which communicate over heterogeneous interfaces, and provides abstraction from the objects as well as offering multiple services [60]. Because of limitations in device capabilities and resources, such as computation power and memory, middleware for the IoT is often located outside of a device's firmware and manages the IoT in a centralized manner (Figure 5).

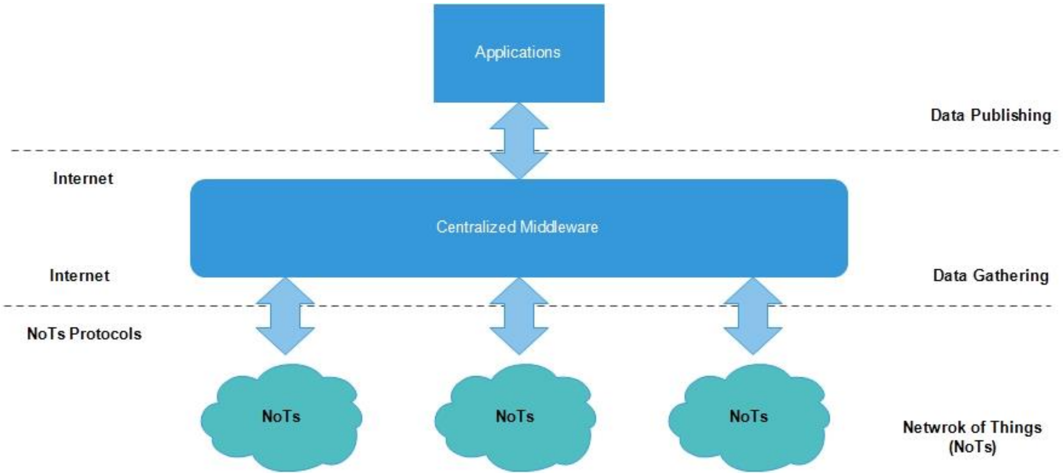


Figure 5. Centralized middleware for IoT

In order to design middleware for the IoT, several important aspects should be considered, the first of which being that centralized middleware needs to abstract the device capability description in a standard way and present it as a universal interface. Secondly, IoT middleware should be responsible for monitoring the state of the devices, managing data collection and providing security and privacy. The important functional components of middleware are explained in [60-62] and [30]. These functional components include, but are not limited to, interoperation, context awareness, device discovery and management, security and privacy, scalability, and data volume management. Interoperation is the ability to retrieve and share information among different application domains that use diverse interfaces. Interoperation can be categorized as: network, syntactic and semantic. Network interoperation, which deals with communication protocols, attempts to define these protocols in order to exchange information. Syntactic interoperation deals with the structure and format of information while semantic interoperation aims to understand the meaning of information and context [60]. Context awareness is the ability to detect the context from raw sensor data and analyze it in order to make decisions [43]. Device discovery and management is the basic functionality for every middleware in order to connect sensors and devices to each other, and discover the services and things capabilities. Managing data volumes is a very important functional component of any IoT middleware. The IoT deals with a large number of sensors and objects, hence the volume of data exchanged between the devices and applications is significantly high.

In [60], the authors surveyed popular middleware solutions, and analyzed and compared them based on characteristics and functionalities such as interoperation, platform portability, context awareness, device management, and security and privacy. These characteristics are typical functionalities of most middleware but, in order to design a general middleware, other functionalities need to be supported, including: scalability; community support; working in different application domains; and Big Data management.

Table 2. Comparison of different IoT middleware (extended version of [60])

Middleware	interoperability	Platform Portability	Context Awareness	Device Management	Security and Privacy	Scalability	Community Support	Ability to work in different application domains	Big Data Management
Hydra	Yes	Yes	Yes	Yes	Yes	**Yes**	**No**	**Partially**	**No**
ISMB	No	Yes	No	Yes	No	**-**	**No**	**-**	**-**
ASPIRE	No	Yes	No	Yes	No	**Yes**	**No**	**No**	**No**
UBIWARE	No	Yes	Yes	Yes	No	**Yes**	**No**	**No**	**No**
UBISOAP	**partially**	Yes	No	Yes	No	**No**	**No**	**No**	**No**
UBIROAD	**partially**	Yes	Yes	Yes	Yes	**No**	**No**	**No**	**No**
GSN	No	Yes	No	Yes	Yes	**No**	**No**	**No**	**No**
SMEPP	No	Yes	Yes	Yes	Yes	**No**	**No**	**No**	**No**
SOCRADES	**No**	Yes	No	Yes	Yes	**No**	**No**	**No**	**No**
SIRENA	**No**	Yes	No	Yes	Yes	**No**	**No**	**Partially**	**No**
WHEREX	Yes	Yes	No	Yes	No	**No**	**No**	**Partially**	**No**
COMPOSE [19]	**Partially**	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**No**	**Yes**
BUTLER [13]	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**Partially**	**No**
DIAT [18]	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**No**
RIMWARE [63]	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**No**

Table 2, an extended version of [60], shows a summary of comparisons in more detail with the extensions in bold and shaded. Scalability is the ability of middleware to handle a growing number of devices and sensors in the sensing layer. Community support, as a functionality of middleware, can collect data not only from individuals but from communities, and can also analyze data at the community level. Smart cities work with many application domains, such as healthcare, transport and public safety, thus the ability to work in different application domains is an important functionality of middleware. Big Data management is the ability to collect, deliver, store and analyze vast amounts of structured and unstructured data. The goal is to ensure a high level of data quality and accessibility.

As Table 2 shows, although the majority of middleware solutions do not support interoperation and context awareness, all of the middleware can support device management, which is essential for connecting sensors to each other and making the IoT feasible. Although some of the above middleware solutions support context awareness, they did not completely fulfil IoT demand and still have some weaknesses. Furthermore, the majority of the above middleware neither support Big Data management nor work in different application domains. Finally, none of the above architecture supports the community concept. As the objective of this present work is to design a general architecture for the smart city, it is very important that the architecture proposed in this research supports all these characteristics, otherwise it will not support diverse smart city applications. The architecture should be scalable in order to handle a growing number of devices and sensors. It should be interoperable to work in different application domains, and support heterogeneous environments as well as Big Data management and context awareness in order to manage the growing amount of data. Most importantly, it should efficiently support communities by managing resources and decreasing the cost of sensing and analyzing.

The following section provides an in-depth review of the most famous platforms in a specific domain, namely healthcare. The review shows that most of the platforms do not support community structure, therefore future IoT platforms can benefit by enabling community structure.

- **IoT platforms designed for healthcare**

Many large companies, such as Microsoft, Google and IBM, as well as many academic institutes, have developed health care applications and platforms. HealthVault was developed by Microsoft in 2007 to aid families and individuals to monitor their health [11]. This software analyzes a family's healthcare, including health history as well as various physiological measures, and accepts data from various sources, such as mobile phones and desktop computers. It also has an open SDK for developers who wish to develop software.

Another example is Google Flu [64], which is based on Big Data analysis tools for detecting and predicting flu in specific geographic areas. Google Flu tries to predict the spread of flu according to Internet searches. If, in a specific location the number of searches related to flu has significantly increased, there is potential that this area has been infected with flu. Google Flu can also track a flu outbreak and predict the next infected area.

Artemis [51], a Big Data analysis framework for neonatal monitoring, employs IBM InfoSphere to process health data in real time by applying temporal data mining techniques patented by McGregor [65] in order to analyze streamed data. Artemis Cloud brings the possibility of supporting and monitoring patients within Intensive Care Units (ICU) in rural areas without the need to transfer them to urban centers.

CHRONIOUS [16], a platform for monitoring and managing patients with chronic diseases, uses a supervised classifier and rule based system to make decisions and determine the severity of a patient's health condition.

LiveNet [66] is a distributed mobile platform for a wearable health monitoring system intended for long-term ambulatory health monitoring and real time processing. LiveNet uses a variety of sensors, such as the Electrocardiogram (ECG), Electromyography (EMG), accelerometer and gyroscope, to collect biomedical data from users. It follows a 3-layer architecture to collect, disseminate and analyze data. LiveNet, which uses real time feature extraction along with context classification, can be used to detect epilepsy seizures, Parkinson's and other chronic diseases.

AMON is an advanced Wearable Health Monitoring System (WHMS) aimed at detecting high risk cardiac/respiratory patients who would be limited to hospital [67]. AMON developers designed this GSM-based secure communication to transfer data from a user device, which is wrist-worn, to a telemedicine center. AMON then processes the data and identifies high risk patients.

AUBADE [68], a platform developed at Ioannina University, Greece, aims to determine the emotional state of an individual. It uses ECG, EMG, and respiration to establish and classify an individual's physiological condition.

Table 3 compares the selected WHMS projects according to cloud support, community support, scalability, interoperability, context awareness and the ability to work on different application domains.

Table 3. Comparison of WHMS

Project	Cloud support	Community support	Ability to work on different applications	Scalability	Interoperability	Context awareness
HealthVault	✓	×	--	✓	✓	✓
Google Flu	✓	✓	×	✓	×	✓
Artemis	✓	×	✓	✓	×	✓
CHRONIOUS	×	×	×	×	×	✓
LiveNet	✓	×	×	✓	×	✓
AMON	×	×	×	×	×	×
AUBADE	×	×	×	×	×	✓

In addition to those previously mentioned in this chapter, there are many other platforms available for WHMS, including WelchAllyn, HeartToGo, Human++ and WiMoCA [69], all of which try to measure an individual's physiological signals and analyze them in order to find abnormality in sensed data. Furthermore, these platforms enable early detection of different medical conditions as well as illness prevention and self-management of chronic diseases. However, none of them work on integrated data sensed from different patients and different communities nor combine health data with other useful information, such as a user's social activities. This ignores an important dimension of healthcare: how individual health is affected by other people. If the system includes information about individual interaction and social activities, it can offer a better perspective on personal health.

2.1.5 Cloud computing and IoT Middleware

A cloud is a large computation and storage resource that is accessible over the Internet. The main strategy behind the cloud is to provide on-demand access to resources and services with the ability to scale usage from just one individual user to millions of users. The cloud is categorized into three different taxonomies according to the services that it can provide. These categories are: Infrastructure as a Service (IaaS); Platform as a Service (PaaS); and Software as a Service (SaaS) [70]. Cloud computing functionalities benefit the development of IoT middleware in several aspects, such as: device accessibility from anywhere without downtime; the ability to collect and store a large amount of data; and powerful computation capability.

The smart city deals with a high number of devices, from smartphones to sensors, which are connected to the Internet. These devices provide various kinds of services and produce a vast amount of data. "Cloud computing is a model for on-demand access to a shared pool of configurable resources that can be easily provisioned as Infrastructure (IaaS), software and applications (SaaS)" [70]. Therefore, cloud computing is an ideal solution for IoT middleware and smart city platforms.

2.1.6 Device Capability Abstraction

The primary goal of device capability abstraction is to provide a common interface to access different objects and things in the IoT. However, because of device diversity and heterogeneity, it is very difficult or even impossible for objects and IoT devices to be described in a universal manner. In order to solve this challenge, two different and opposite methods, are available [63]: external descriptive and self-descriptive.

- **External Descriptive**

The external descriptive solution for device capability abstraction uses an agent, such as a middleware component, to provide a universal interface to access different types of devices. This solution is often used in SOA based architecture with device description exposed as web services. This solution has limitations since agents such as middleware components require knowledge of a device's capability. Also, the agent should be reconfigured each time a new device attempts to connect to it. The weakness of an external descriptive solution is that it suffers from lack of scalability and security.

- **Self-descriptive**

In a self-descriptive solution, no agent is needed to abstract the device capability. Instead, the device itself provides the capability description so it can be directly accessible. In this method, the device capability description is stored on the device and can be retrieved through a communication protocol. The device can then directly share its capability with others without assistance from an external entity such as middleware. The most important benefits of this solution are interoperability and automatic configuration.

As this present work proposes to design an interoperable architecture to support different application domains for smart cities, using the self-descriptive method for device capability abstraction is one of the ideal solutions.

2.1.7 Context-aware Computing for the IoT

The concept of context-aware systems was introduced by Weiser [71] in 1991 for ubiquitous computing and then, in 1994, Schilit and Theimer [72] used this same term in their research paper. Subsequently, *context* and *context awareness* were used in many research studies. There are many definitions for context, such as the definition by [73] which considered context as the five Ws (Who, Where, What, When and Why) or the definition by [74] who claimed that “the context is any kind of information that can be used to characterise the situation”. Furthermore, as the term implies, the context awareness system uses context in order to provide relevant information to its users [74]. Typical context-aware systems can support acquisition, representation, delivery and reaction [75].

Due to the number of sensors and the large volume of information which is produced by these sensors in the IoT, context awareness plays a crucial role in deciding what data should be processed and in identifying the degree of awareness. Data that is collected from sensors is worthless unless processed, analyzed and made understandable [43]. Therefore, context-aware computing allows the context information linked to the sensors’ data to be saved, thus enabling more straightforward analysis.

Traditional methods, such as directly connecting the sensors to the application, are not feasible for the IoT because it works with billions of sensors which generate a vast amount of information. In order to tackle this inefficiency, the context-aware middleware solution was proposed in many research studies, including the work conducted by [76-78]. In addition to the mentioned research studies, a review of relevant literature reveals surveys which consider various aspects of context awareness. As an example, in [79], the author reviewed context representation and context reasoning. In [80], the authors focused more on context modeling and reasoning techniques. Context-aware architectures were also surveyed in [81].

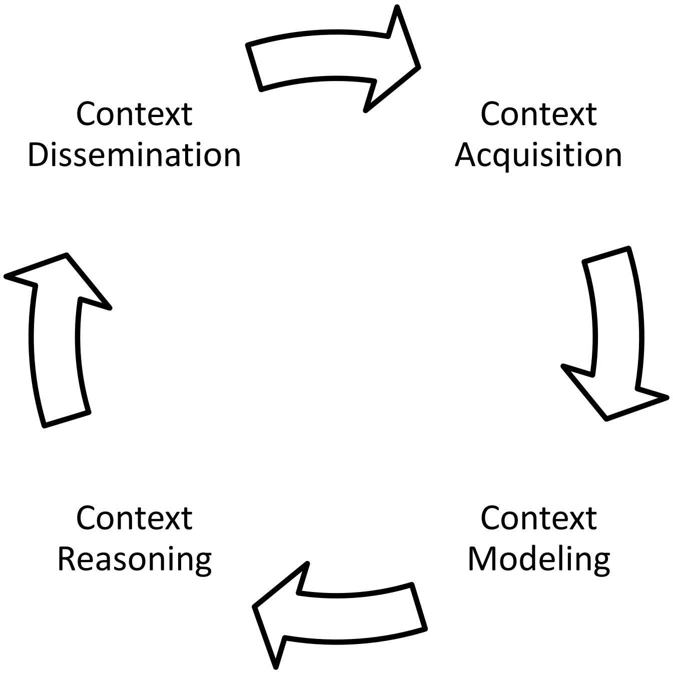


Figure 6. Context life cycle

The authors in [43] identified that the typical context management system has four phases: context acquisition, context modeling, context reasoning and context dissemination (Figure 6). In order to have a scalable, interoperable, secure and cost-effective architecture, all these phases should be considered. Moreover, enabling the community structure for the proposed architecture requires employing a context-aware system to convert raw community data into meaningful information.

2.1.7.1 Context Acquisition

Context acquisition is the initial step for any context-aware system. Many techniques are available for context acquisition; such techniques vary according to context source, context type, frequency, responsibility and acquisition process [43]. Figure 7 shows these general context acquisition techniques in more detail.

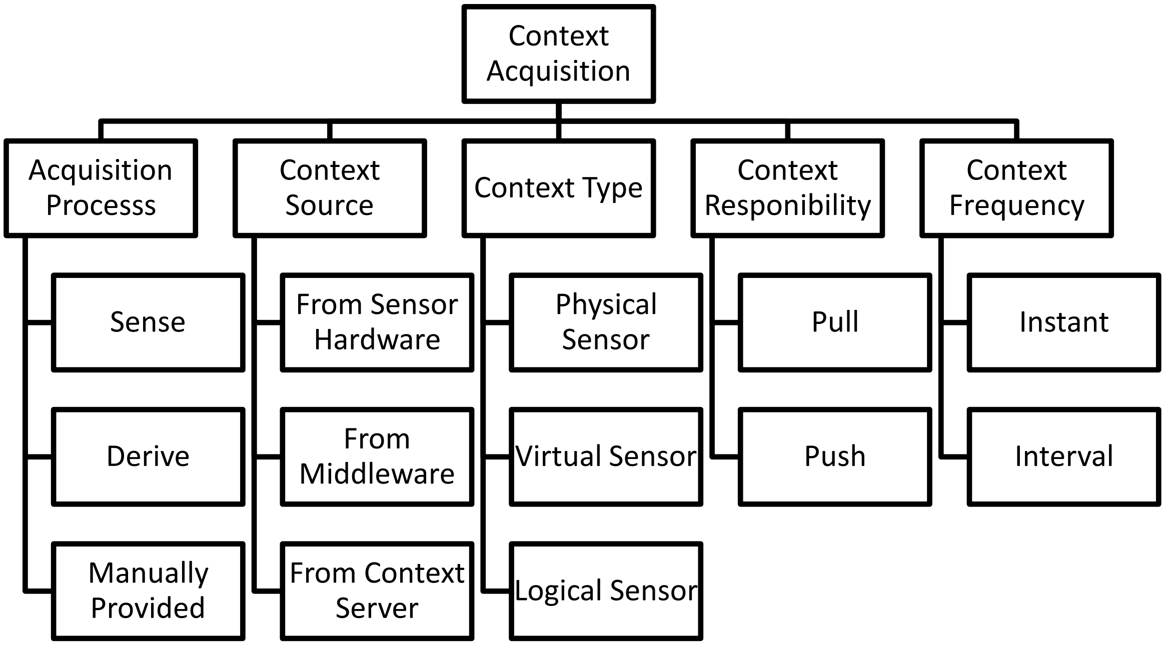


Figure 7. Context acquisition techniques

Context acquisition can be performed by two methods: pull or push. If the pull method is used, the application is responsible for gathering data from sensors and communication. In contrast, in the push method the sensors are responsible for sensing the environment and sending data to the upper layers. The context can be directly acquired from sensor hardware, middleware or from context servers which store the context.

2.1.7.2 Context Modeling

Several modeling methods are presented in the literature. In [78] and [82], the authors introduced some popular methods, with their respective advantages and disadvantages, that are used by many context-aware systems. Table 4 compares the available methods based on validation, scalability, flexibility, and standardization and application independency [43, 82]. Ontology-based modeling seems a suitable candidate for this research study because it is scalable, application independent, and interoperable, and has the ability to support complex structures.

Table 4. Comparison between context modeling methods

Modeling Methods	**Key-Value**	**Markup Schemes**	**Graphical**	**Object based**	**Logic based**	**Ontology based**
Validation	×	✓	✓	×	×	✓
Scalability	×	×	×	×	✓	✓
Flexibility	✓	✓	✓	×	✓	×
Processing tools	×	✓	✓	✓	✓	✓
Application independency	×	×	✓	✓	×	✓
Standardization	×	×	×	×	×	✓

2.1.7.3 Context Reasoning

Context reasoning is the means for changing data into knowledge in order to better understand the data [83]. Context reasoning consists of three different steps: preprocessing, data fusion and context inference (Figure 8)[43].

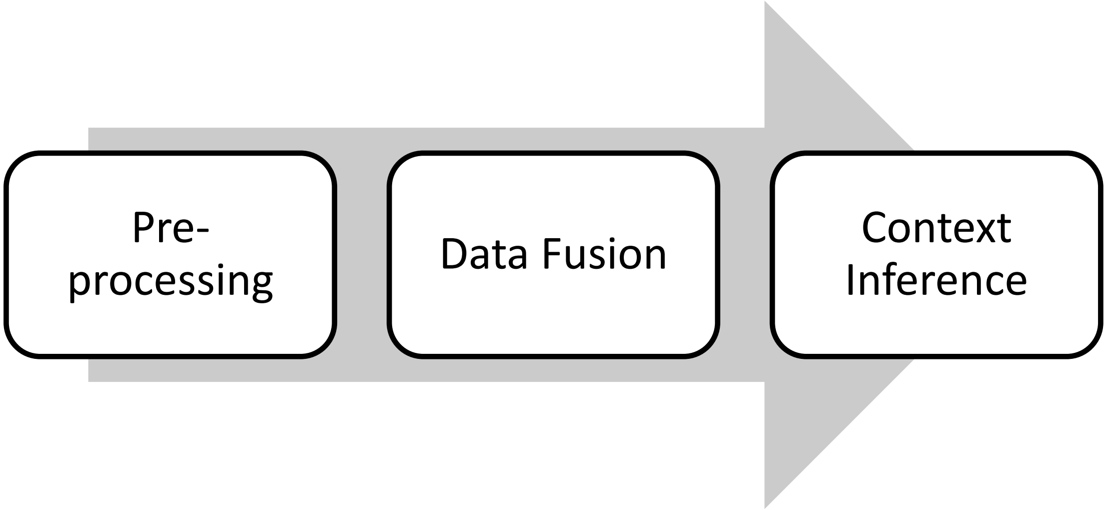


Figure 8. Context reasoning steps

Context reasoning is classified into six distinct categories, as depicted in Figure 9.

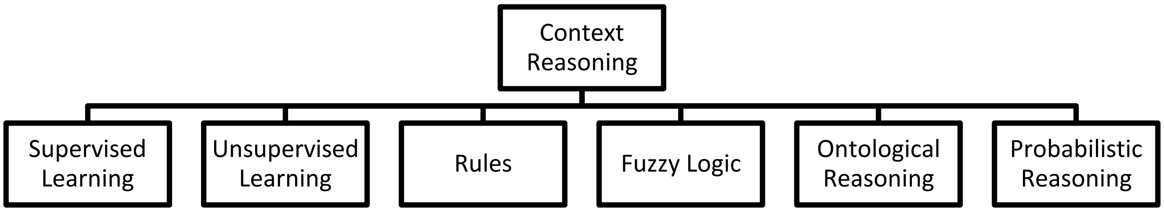


Figure 9. Context reasoning techniques

Table 5 compares context reasoning techniques and highlights the main pros and cons [43].

Table 5. Pros and cons of context reasoning techniques [43]

Techniques	Pros	Cons
Supervised Learning	• Is accurate • Has mathematical and statistical foundation	• Requires significant amount of data • Every data element needs to be converted into numerical values • Selecting feature set could be challenging • Can be more resource intensive (processing, storage, time) • Less semantic, hence less meaningful • Training data is required • Models can be complex
Unsupervised Learning	• No training data required • No need to know possible outcome	• Models can be complex • Less semantic, hence less meaningful • Difficult to validate • Outcome is not predictable • Can be more resource intensive
Rules	• Simple to define • Easy to extend	• Needs to manually be defined • Can be error prone due to manual work • No validation or quality checking
Fuzzy Logic	• Simple to define • Easy to extend • Can handle uncertainty	• Needs to manually defined • Can be error prone due to manual work • No validation or quality checking
Ontology Reasoning	• Allows complex reasoning • Allows complex representation • Validation is possible	• Data needs to be modelled in a compatible format (e.g. OWL, RDF) • Limited numerical reasoning
Probabilistic Reasoning	• Allows evidence to be combined • Can handle unseen situations • Alternative models are available • Can handle uncertainty • Provides moderately meaningful results	• Should know the probabilities • Analyzes numerical values only

2.1.7.4 Context Dissemination

Context dissemination is the way that consumers acquire data or context information. There are two well-known methods available in the literature for context dissemination: query and subscription [43]. In the query method, users ask for a specific context from the context management system and their requested information will be provided by accessing sensor hardware or by composing and integrating multiple contexts. In the subscription method, users register themselves for an event or for specific sensor(s) data and the context management system will update the user when the event occurs.

- **Publish subscriber model:**

The publish subscribe model [84] is broadly used in middleware, large scale information dissemination and enterprise applications. The main goal of the publish/subscribe model is to exchange information between publishers and subscribers. Messages are transmitted by publishers and show subscriber interest in the status of the system. Subscribers also directly submit their interest to the system and will receive notification regarding desired publications. Neither publishers nor subscribers know about each other and only submit their message or interest to the underlying system, known as the messaging fabric. Figure 10 shows the publish/subscribe communication paradigm for this model.

Both publishers and subscribers can communicate with only one entity, which is the messaging fabric. The messaging fabric saves all subscriptions associated with respective subscribers. Participants in this model do not need to know each other as the scale of the system grows. Hence, the system is totally scalable.

The publish/subscribe model is also known as the event filtering and matching approach. The subscriber represents the filter and the publisher collects the events or observations, while the message fabric makes an association between the events and subscribers, based on their subscriptions.

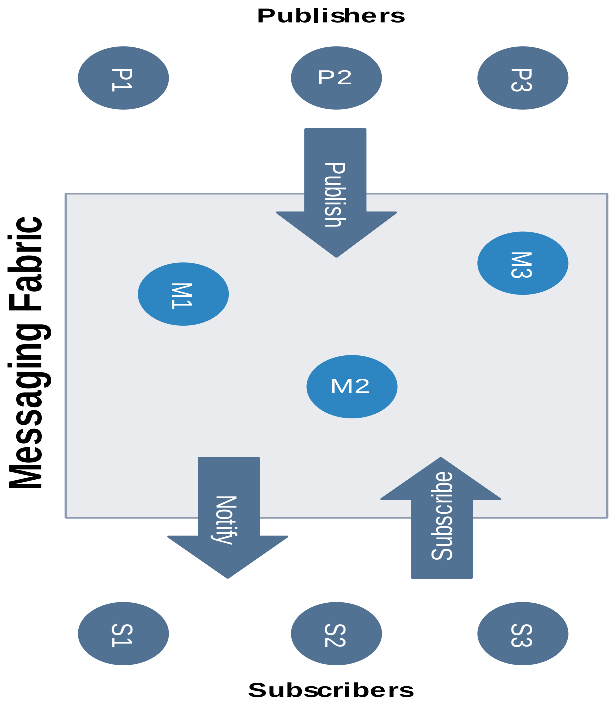


Figure 10. Publish/Subscribe model

Several subscription models are available in the literature [84, 85], and are characterized by different expressive powers. However, the topic-based and content-based publish/subscribe are the two best known models. As an event filter, the topic-based approach filters publications according to topics associated with a message while the content-based approach filters out publications based on their content. For both, topic and content-based, the types of topics, or semantics of content, to publish or subscribe, are either out of band information and must be known to clients, or are dynamically discoverable by clients based on additional support provided by the systems. Publish subscriber is an ideal solution for increasing the scalability of the system, therefore it can be used as the content dissemination technique for the desired architecture.

2.2 Community Sensing and Mining (CSM)

As a result of progress in ICT, individuals are engaging in and connecting via different forms of communities, such as social networks and cyber physical networks. Involvement in community is how people interact and share content with each other. Therefore, monitoring urban and community dynamics leads to better city management and urban planning [32] and provides information for decision making [24]. Since the last decade, social networks and cyber-physical space have gained in popularity. Social networks allow people to share digital content such as pictures, music, and videos, and to communicate with each other. Cyber physical space uses opportunistic contacting and ad-hoc connection among individuals as well as between pairs of devices, such as mobile phones and vehicles.

2.2.1 Community Definition

The word "community" has been widely used in different research studies. A community is a group of people or objects with common characteristics or similar interests, that are tightly connected via various social and physical processes. A review of the available literature confirms that social studies was the first context to extensively use the term community [86]. The Oxford English Dictionary defines community as "the people of a district or country considered collectively, especially in the context of social values and responsibilities". The idea of community is tightly associated with the term "network", meaning connectivity and interactions. With recent progress in ICT, individuals are engaging in and connecting via different forms of community, such as social networks and cyber physical networks. Involvement in community is how people interact and share content with each other. Therefore, monitoring urban and community dynamics leads to better city management and urban planning [32]. By building communities in the smart city, the environment can be sensed beyond a single individual ability, life quality can be increased and the cost of sensing can be reduced for all community members.

A community in ICT can be a form of online, offline or cross-space community [87]. In an online community, people interact with each other by sharing content using ICT. Offline communities are formed opportunistically by co-located people during their daily activities. A cross-space community integrates both online and offline communities. Cross community sensing and mining focuses on the interaction among different and heterogeneous communities, and emphasizes the association and aggregation of multimodal data, which is obtained from distinct communities. Communities can also be formed by connecting objects that may have common interests and which can interact and collaborate with each other.

2.2.2 Mobile-based Community and Crowd Sensing

Mobile-based Community and Crowd Sensing (MCCS) is a new method that collects information from individual-companioned devices such as smartphones, smart vehicles and wearable devices [24]. This method uses the power of the crowd to sense, analyze and share local knowledge, which is obtained from individual mobile sensor devices [32]. The obtained data can later be aggregated and fused in order to mine further knowledge or crowd intelligence extraction. The aggregation of sensed data from a group of individuals can result in better decisions compared to sensed data obtained from a single user.

2.2.3 Community Sensing Characteristics

The following section lists the most important elements of community sensing, together with a brief explanation.

- **Data**: The data collected from communities is heterogeneous and multimodal. This data is collected via virtual or physical communities. Different properties of individuals can be extracted from raw data collected from the different communities. With the rapid development of social networks, a vast amount of data is produced by individuals in virtual communities, which can be linked to their mobile sensed data to monitor city and community dynamics.
- **Sensing style:** The data sensed for CSM is classified into two distinct categories: explicit and implicit [88]. Explicit data collection is the data collected from mobile devices, which is the main purpose of the applications and individuals who participate in sensing tasks. In contrast, the data collected from social networks and virtual communities is implicit data collection because the main goal of participants is social interaction rather than data collection.
- **Technologies**: The most important technologies in CSM are mobile sensing, community analysis and data mining.
- **Applications**: Applications in community sensing can be varied, from recommendation systems to targeted advertising.

2.2.4 Community Analysis

Since the last decade, there has been tremendous growth in online social services. People share content via email, instant messaging, and social networks. More recently, as the Internet has expanded into the new era of the IoT, analyzing the data from the interaction of people and things in online or physical communities has many benefits. This analysis can allow people to be more involved in communities and to benefit from each other, as well as help communities to better manage resources and infrastructure.

2.2.5 Community sensing applications

Community sensing applications can be divided into three distinct categories: environmental, social and infrastructure [88]. Environmental community sensing deals with natural phenomena, such as disasters, pollution or the water level in creeks. An example of an application in this category is CommonSense [38], which uses handheld mobile devices to measure air pollution. These devices, when distributed among a large population, can collectively measure pollutants in a large-scale area. Infrastructure community sensing deals with city infrastructure, such as roads and traffic. An example of an application in this category is CarTel [89], which uses installed sensors on cars to measure speed, velocity and traffic. The final category of community sensing is social community sensing, where individuals share sensed data and benefit from the collective data. An example of an application in this category is DietSense [34], where individuals take a picture of their food and send it to their chosen community in order to compare eating habits.

2.2.6 Cross Community Mining

Cross community sensing and mining focuses on the interaction among different and heterogeneous communities, and emphasizes the association and aggregation of the multimodal data obtained from distinct communities.

2.3 Implications for this Research

Progress toward a community aware sensing system requires a generalized architecture and more community structure to sense and analyze data, at both individual and community levels. The requirements of a general reference architecture, which supports most of the above-mentioned characteristics such as scalability, interoperability, context awareness, Big Data management and community awareness, is an open research area. This system should also ensure security and citizen privacy. Community analytics should support environmental, social and infrastructure perspectives.

Chapter 3

3 Proposed Architecture for the Community-Oriented Smart City

This chapter introduces the proposed community-oriented architecture for a smart city together with its associated framework. The design of this architecture is the starting point for community sensing and mining research, which are the key functional blocks of a smart city. The designed architecture and its associated framework can solve some of the challenges previously mentioned in Chapter 1.

3.1 Proposed Architecture

An architecture is an abstract design concept of an application that shows the relation between different parts and how they are connected. Developing an architecture usually depends on the applications that will use the platform, but designing a general architecture that can fulfill the requirements for all the applications and services in a smart city is challenging. Creating an architecture for a smart city, while considering community sensing and IoT, is a complicated task, mainly because of the exceedingly large diversity of objects and devices, link layer technologies, and services that may be associated with such a system. There is also a high degree of interdependency between the various infrastructures of a smart city, which adds to the complexity of community data analysis [2].

The IoT consists of many "things" such as smartphones, tablets, cameras, and sensors. The number of connected objects increases at a daily rate and the methods of collecting information among the objects change. Consequently, objects produce a vast amount of data and information. Cloud computing is an ideal technique for storing and analyzing this volume of information, as well as for running many services. In the proposed architecture (Figure 11), cloud computing is used to solve some of the challenges mentioned in Chapter 1, including Big Data management. Cloud computing provides access, at any time, from any location, to all resources, such as mobile devices, sensors, actuators, and tags.

The proposed architecture (Figure 11) consists of four layers: sensing, transmission, storage and control, and application.

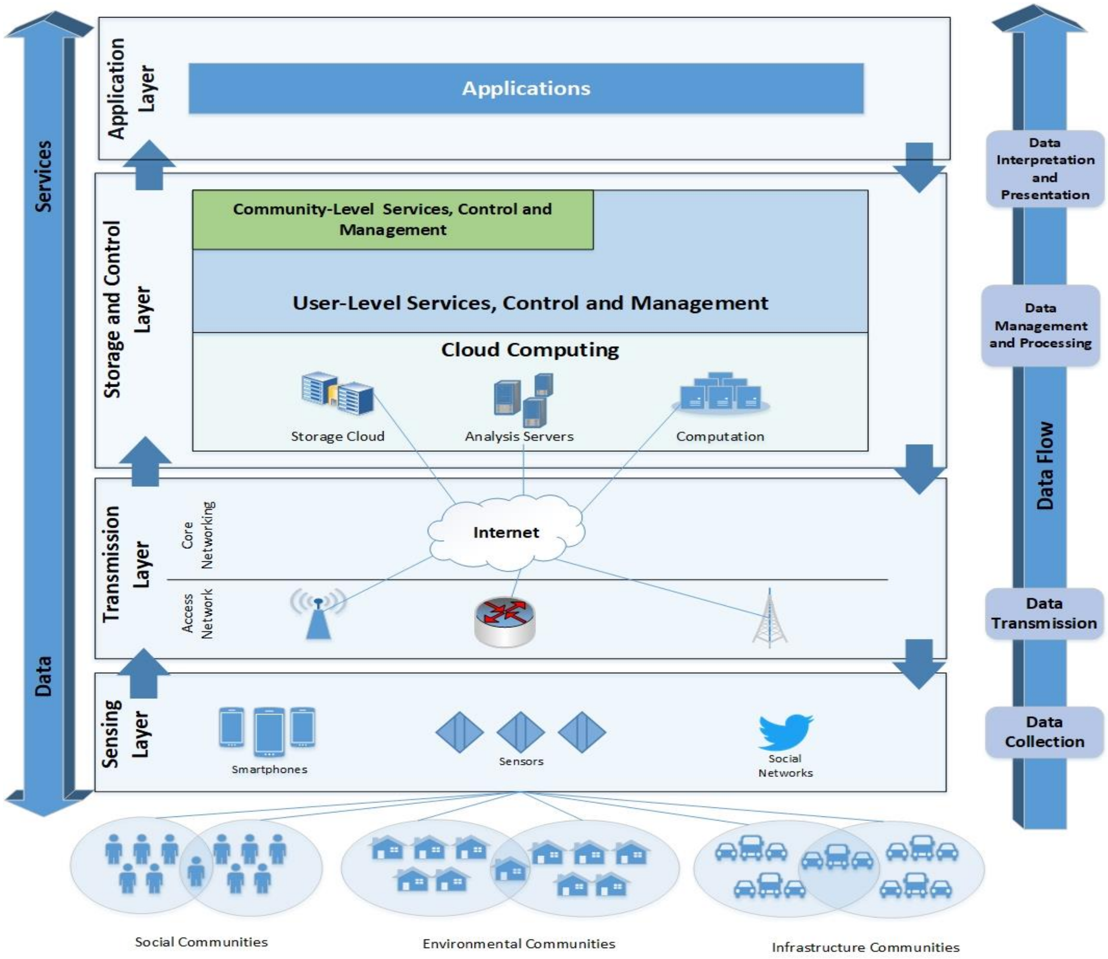


Figure 11. Proposed Community-oriented architecture for the smart city

3.1.1 Sensing Layer and Data Collection

This layer contains different data sources for community sensing, including smartphones, sensors, and vehicles. Heterogeneity is one of the important characteristics of the sensing layer, which often contains a variety of sub-networks that adopt different communication technologies [90]. To overcome the difficulty of collecting data in heterogeneous networks, a generalized framework for data collection is required. This framework should retrieve data, either continuously or at random intervals. The objects in this layer are small and most of them have a limitation on computation and energy. Therefore, it is crucial that data collection algorithms and techniques are designed to efficiently use energy.

There are three sensing resources in the proposed architecture: WSNs, mobile devices and social networks. Since they enable the collection, processing and analysis of data in any kind of environment, they play a crucial role in the sensing layer. Based on the expansion of social networks and the increasing number of mobile devices, citizens share their social activities in interactive environments [91]. With this participatory sensing, service providers can collect data more easily than with other methods.

3.1.2 Transmission Layer

The transmission layer includes a communication infrastructure that delivers the data from the sensing layer to the control layer, and vice versa. As most of the data sources in the proposed architecture are mobile devices, depending on the data collection environment, there may be no network infrastructure available to transmit the data to the destination. In this case, it is the responsibility of the transmission layer to deliver the data to its destination in an opportunistic manner. As the aim is to route the data through the

Internet, the entire platform needs to adapt to IP. Many alliances and councils have tried to adapt the IoT to IP technology. For instance, 6LoWPAN [92] and Internet0 have followed the approach of reducing the complexity of an IP stack to achieve a protocol designed to route IP over everything [9]. In the proposed architecture, the focus is on how to connect objects to each other for information exchange rather than enabling IP technology to things and objects. For this purpose, using a gateway is considered as an interface between the sensing layer and core transmission layer to translate the protocols used in the sensing layer to IP.

3.1.3 Control Layer

This layer is responsible for: retrieving data from a data base; applying data mining algorithms to find patterns in the data; registering and managing the services that are provided by multiple service providers; managing communities; allocating tasks; and processing crowd data. To achieve these goals, a powerful computational resource is required, hence a cloud based analytical and computational module is considered in the proposed architecture. All data that is collected from the sensing layer is processed on the fly on its way to the database. The control center does not need to individually communicate with each entity or sensor to obtain the data.

In the proposed architecture, a distinction is made between community services and individual services, therefore this layer is separated into two control centers. To communicate with other entities, such as remote users or the monitoring section, the web interface may be considered as a general interface for control centers. Each control center consists of database management, knowledge discovery and service management components. The notable difference between a community control center and an individual control center is the type of knowledge discovery methods applied to the data. Furthermore, service providers at the community level are more interested in providing a service for the community than for individuals.

The proposed architecture allows developers, service providers and data miners to join a network and offer their services.

- **Community-Oriented Services**

The main objective of this research is to design a community-oriented architecture for smart cities to efficiently manage resources and improve the quality of life for citizens. In smart cities, a large number of objects communicate over the Internet and many users tend to access their data at the same time. Therefore, this interaction will result in a steady increase in network traffic. The current Internet architecture, which is designed to handle host-centric communication, does not support data-centric communication [23].This difference between the current architecture and the new service model architecture causes a waste of network resources and increases the communication overhead. Furthermore, it dramatically decreases the performance of the whole network.

Categorizing smart city objects into groups based on their common characteristics is a solution for the above problems. It is possible to create communities and group the users and objects into the same community, according to their interests. This solution decreases the communication overhead and ignores unnecessary communication to increase the total performance. An additional advantage of this integration is better service performance and resource discovery. Furthermore, it guarantees scalability and increases the level of trustworthiness by interacting with more objects [93]. Considering the sets of objects in communities is also a kind of collective intelligence, which brings benefits for all community members. Figure 12 shows the initial concept of community-based services.

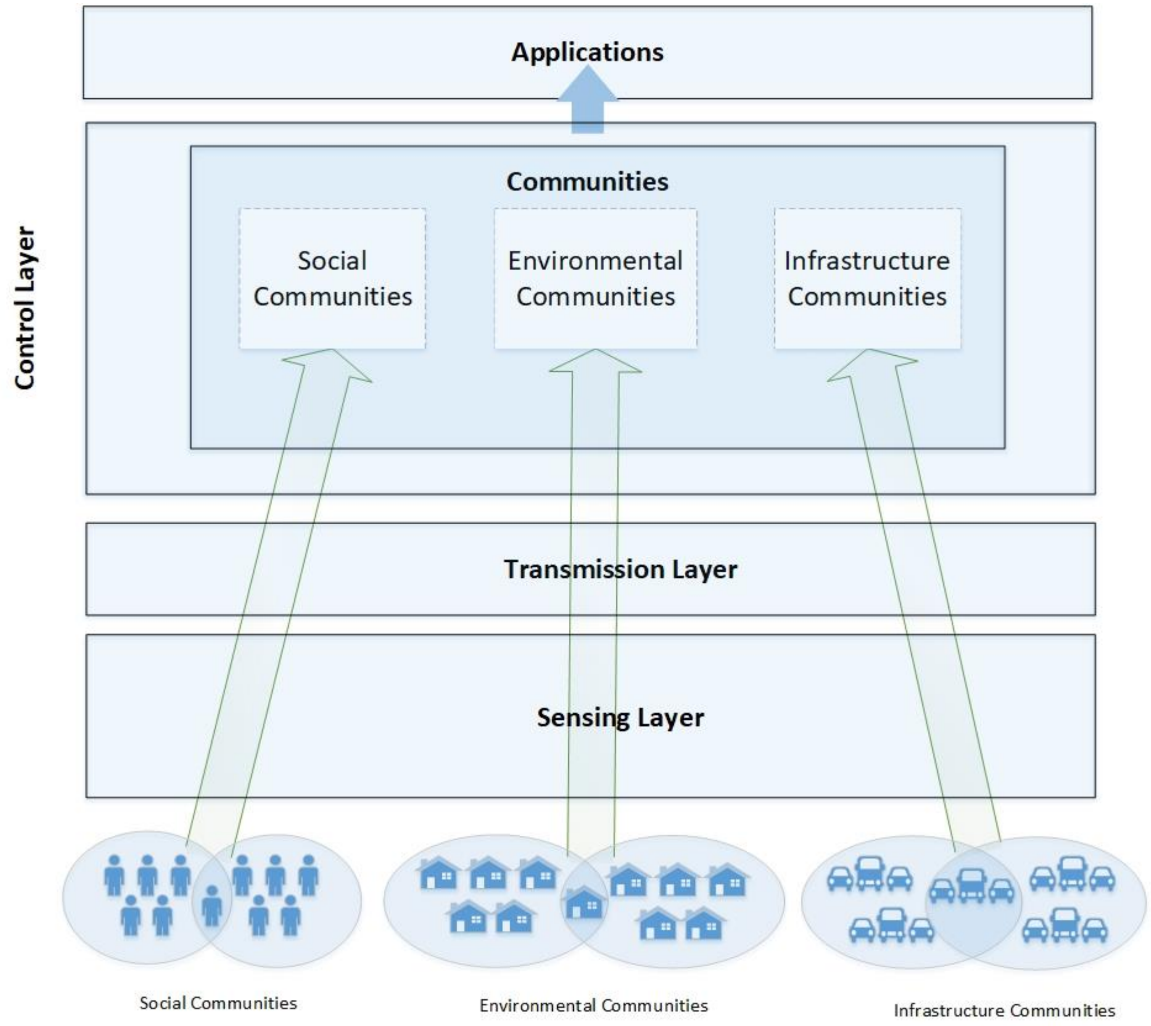


Figure 12. Community-based Smart city architecture

With the concept of a community-based smart city, objects can be associated with the service they can deliver. Therefore, how to publish the information and services as well as how to find the appropriate service in the communities are key issues. By enabling community structure for smart city objects, users can discover services by searching among the objects that have a common interest instead of relying on Internet discovery tools, which are not scalable for a large number of devices.

The object relationship in the community-based smart city can be categorized into multiple taxonomies, such as: (1) Objects that are co-located and co-operate for the common goal that is associated with the location; (2) Co-work objects that collaborate with each other in order to provide a common smart city application, such as wearable sensors that work together to monitor health; and (3) Social objects that connect to each other because the owners of the objects have a relationship with each other. Objects can also belong to a community according to how willing the members are to either share the information or subscribe in order to obtain the information in a specific title. For example, all wearable sensors, which belong to one person and that work together to monitor the user's health, can join with the city's health community to share their information.

3.1.4 Application Layer

This layer includes different types of applications and services for individual and community sensing and mining. This layer is also responsible for displaying and visualizing the results that are collected and mined by lower layers. Another functionality of this layer is to provide an interface for users to interact with the rest of the system.

3.2 Framework for Community-Oriented Smart City

A framework, which is an implementation of the selected architecture, indicates the different components that need to be implemented in order to make the architecture feasible. The proposed framework for community-based smart cities, which is derived from the main architecture, has four different layers: sensing, network (data transmission), control and application. Figure 13 shows the four different layers included in the design of the framework for a community-oriented smart city. The control layer consists of three sub-layers: community analysis, data management and device management.

This section provides a more detailed analysis of all the components of the community analysis, data management and device management sub-layers. These components are presented below in Figure 13:

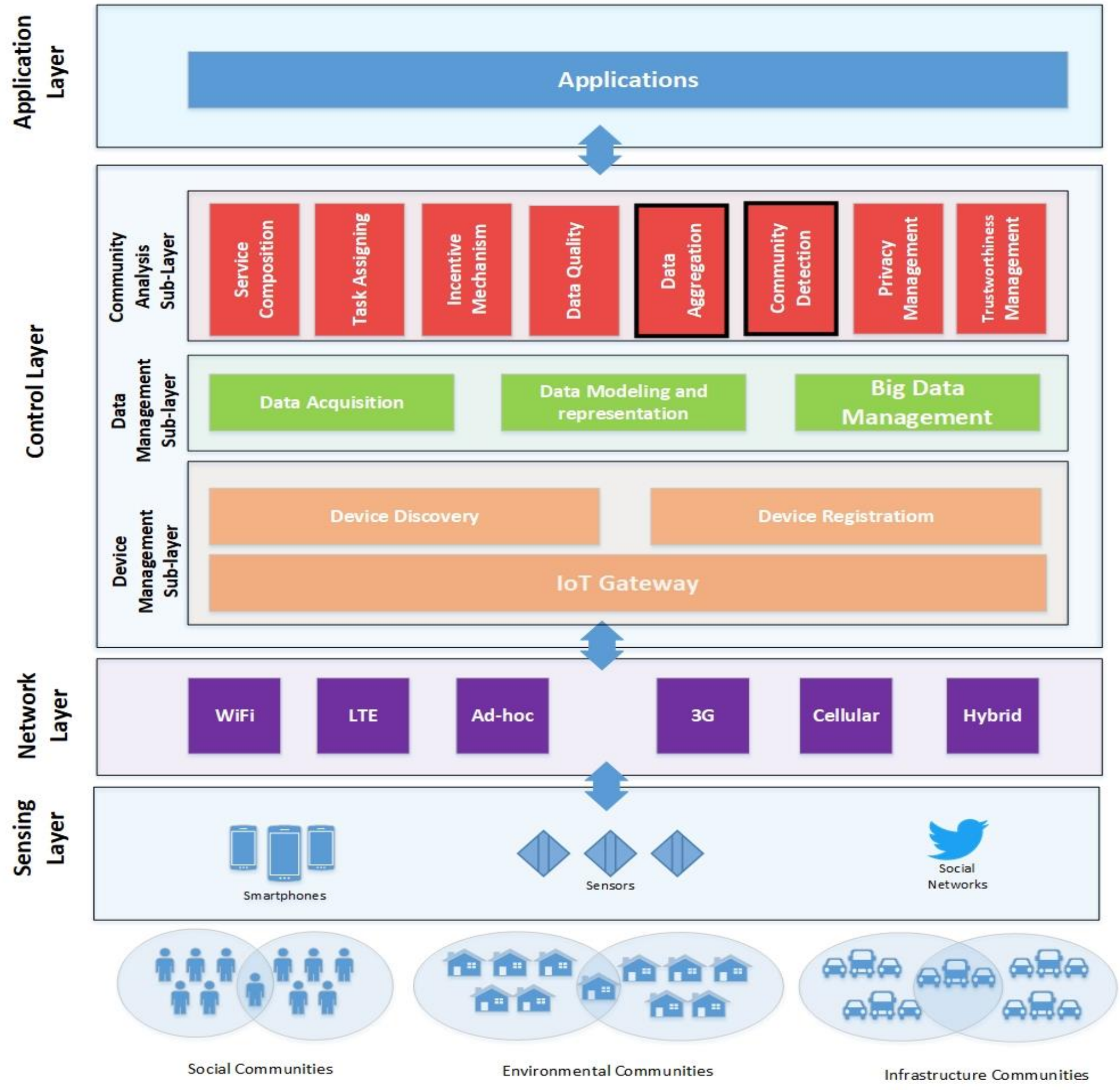


Figure 13. **Conceptual framework for the community-oriented smart city**

- **IoT Gateway**

The IoT Gateway component provides a unified interface for the upper layers and components to access sensor data and IoT objects without the problem of dealing with various kinds of sensors and different sources of information [94]. The IoT gateway solves the heterogeneity problem mentioned earlier in Chapter 1 [94]. It also solves the scalability problem because new devices and objects can be easily added to the system. An IoT gateway can use both external descriptive and self-descriptive solutions [63] to provide a transparent view of the sensing layer to the upper layers. With the emergence of fog computing [95], the IoT gateway can play an important role in extending the cloud computing paradigm to the edge of the network. Each community can be associated with separate IoT gateways and different tasks, including storage, and preprocessing can be efficiently completed from the community perspective.

- **Device Discovery**

The device discovery component is responsible for identifying the appropriate devices to accomplish the tasks assigned by the upper layers [96]. Since a vast number of devices and sensors connect together [29], the existence of this component in the proposed framework is vital. Once the appropriate device with proper characteristics for the sensing task is found, a sensing task can be assigned to it. Several solutions have been proposed in the literature to solve the device discovery problem [96-99]. Some of them are focused on single characteristics such as service description and communication protocols while others are focused on multiple specifications. As a smart city is expected to host millions of devices, the two important factors for the device discovery service are the reliability of discovery and response time. Device discovery can also be performed in two different ways: centralized or decentralized.

- **Device Registration**

The device registration component is responsible for automatically registering any device that would like to join the network. This component can extract the device capabilities and create an appropriate profile for each device [63]. Other components such as device discovery can use registered device profiles in order to find appropriate devices for sensing and task assigning.

- **Data Acquisition**

This component is responsible for acquiring data from the sensing layer by applying different acquisition methods, such as the pull or push [43]. The data acquisition component can automatically detect the method for context acquisition based on context type, context source, frequency and responsibility. Different applications may require different data acquisition techniques to acquire data, therefore this component is responsible for detecting and applying proper techniques for data acquisition. This component can also acquire data from different communities. To this purpose, the data acquisition component should work directly with the community detection component in order to access different communities and acquire data from them.

- **Data Modeling and Representation**

The data type in mobile community sensing can be categorized into two distinct groups: (1) User personal/private data, such as health, location or social media and (2) Public data that is also available to other users to sense, such as traffic-related, environmental, and weather data. Since time is one of the common dimensions in all sensor data, this present work considers data as a multi-dimensional time series. If N is assumed as the number of participants, K as the number of dimensions and T as the period of data collection in community sensing, then data can be denoted by $D(d, t, k)$ where $d \in [1, N], t \in [1, T]$ and $k \in [1, K]$. To denote all of the data, a three dimensional $(N * T * K)$ matrix is used, where d_{ijz} is the data of user i in time j for sensor k.

- *Spatio-temporal correlation*

Data that is collected from citizens by mobile community sensing has a spatio-temporal correlation, meaning that the data collected from the same user in different timeslots are correlated if these timeslots are adjacent. Also, the data from different users is correlated if collected at the same periods of time and if the users are physically correlated.

- *False Data Detection*

As previously mentioned in this chapter, collected data may contain missing and false data. False data can be intentionally generated by malicious users or accidentally produced as a result of hardware failure or a noisy sensing environment. One technique that can be used for detecting false or missing data is to build a new data matrix from the original collected database on a spatio-temporal correlation and then compare the new matrix with the original. Figure 14 shows the proposed model and steps for false and missing data detection.

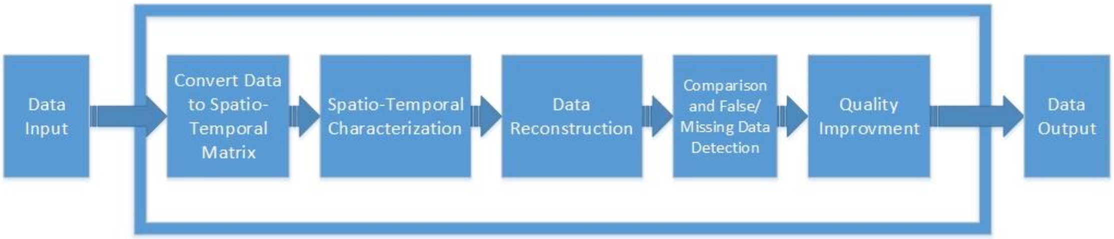

Data Input
Convert Data to Spatio-Temporal Matrix
Spatio-Temporal Characterization
Data Reconstruction
Comparison and False/ Missing Data Detection
Quality Improvment
Data Output

Figure 14. Proposed model for false and missing data detection

The important part of the above model is how to reconstruct the data to approximate, as closely as possible, the real sensed values. In order to precisely reconstruct the original data values, it is necessary to trust the collected data of participants and approximate the original data values by interpolating their respective sensor value. The current literature offers multiple techniques, such as Spatio-Temporal Compressive Sensing [100], Nearest Neighbor and Delaunay Triangulation [101], that can be used to rebuild data matrices.

- Big Data Management

The Big Data management component is responsible for managing and storing the data collected from communities. As the volume of collected data is large, processing and storing it by using existing data management tools is inefficient [11]. Furthermore, because of the variety and multi-modality of the collected data, traditional data management techniques are insufficient [11]. To solve this problem, the Big Data management component transforms and represents data in a uniform manner, based on the same ontology.

- Service Composition

This component is also responsible for composing the different services and creating a new service. A complex service may require mixing multiple services together and composing a new service. Service composition can be accomplished in two different ways: Flow-based and AI-based [63]. In flow-based service composition, the sequence of services that need to run in order to compose a new service is represented by a graph and manually defined. AI-based service composition automatically determines the service needed for composition and dynamically adapts itself by checking the service availability [63]. In a community-oriented framework, services can be associated with communities and can be combined in order to create a large inter-community service. Composition of services is used when the current available services cannot fulfill a task. In such a case, the service composition component is able to explore available services and combine them together to fulfill a request. The service composition component in the community analysis sub-layer needs to work alongside the community detection component to be aware of detected communities and services associated with communities.

- Task Allocation

This component is responsible for taking the application layer requests and assigning them to the appropriate sensors in order to fulfil the requests. The assignment should be based on application or service requirements, such as sampling location, sampling time, device capability, willingness, and given budget [102]. As many users participate in a sensing task, an appropriate task allocation is required to determine which node is suitable for the sensing task.

- Incentive Mechanism

This component is responsible for providing incentive mechanisms and controlling data contributors [103]. In community sensing, as humans are involved and sensors are usually possessed by different individuals, participation in the sensing process depends on individual incentive, and incurs both cost and energy. Without strong incentive mechanisms, participants may not be involved in the sensing process. Therefore, identifying a means by which to motivate individuals to be active in the sensing process is another challenge. Furthermore, if money is used in the incentive mechanisms, participants may be more likely to deceive the system in order to obtain more benefit from it.

- **Data Quality**

This component is responsible for controlling the quality of contributed data, while ignoring low quality data.

Mobile crowd sensing [88], also known as community sensing, enables a vast data collection from citizens while allowing a wide variety of sensors and data sources to contribute data. However, collecting data from multiple sources and different users is often loosely controlled, resulting in outliers, noise, and missing information. To increase the quality of collected data, a data quality component is suggested in the architecture.

The data contributed by communities or a crowd is not always reliable. Users could submit false data to deceive the system in order to obtain more rewards without performing the appropriate sensing task. As examples, a renting agency might contribute fake data to a noise monitoring system to promote a rental apartment in a specific region or an individual involved in a crime may submit fake data to a crime monitoring system in order to avoid police detection.

Low quality collected data is not always attributable to fake data. Other key factors which decrease data quality include sensor hardware failure, communication errors, inappropriate sensing or false sensor readings. For example, in a location-based application, a GPS sensor may submit incorrect location coordinates due to poor GPS signal quality, or data may drop in a wireless communication network as a result of a low quality wireless connection.

Different techniques for tackling data quality have been developed, such as the work introduced in [104, 105]. However, none of them has been presented as a general approach. Different sensors, applications and environments need different approaches in order to resolve data quality. As an example, the technique used for resolving data quality in a GPS sensor cannot be used for a temperature sensor.

- Community Detection

The community detection component is responsible for managing objects and people in communities, recognizing the communities and for identifying the relationship between the objects and people in communities. Furthermore, this component is responsible for analyzing the community and crowd data, and for extracting patterns from raw sensory data by leveraging community data mining techniques. Also, the community detection component enables people to be placed in communities based on attributes that they have as well as through community establishment. This component, together with the data aggregation component, can determine similarities between people by leveraging the sensor data collected from their smartphones or from their surrounding environment and aggregate sensor data for community detection and extraction. The component, which is also able to transform time-series data to similarity networks for community detection, will be discussed in greater detail in Chapters 4 and 5.

- **Data Aggregation**

In the proposed framework, data can be collected from social or physical communities. The data aggregation component is responsible for cross space data association and fusion. In community detection, data aggregation is one of the most important components. Because different types of data have different characteristics and features, data aggregation can fuse data from different sources of information [106].

- **Privacy Management**

Privacy management is designed to maintain the privacy of community sensing participants. Data collected from communities usually contains personal data such as locations and names [106]. This component tries to apply different techniques, such as anonymizing data, to maintain user privacy. Simple anonymization is not always sufficient to preserve privacy [107], especially for data that contains GPS information and locations since the data itself can disclose the identity of the owner even if collected anonymously [108]. Therefore, this component uses different type of techniques and mixes them together to preserve privacy before the data is published.

- **Trustworthiness Management**

  In community sensing, human involvement in the sensing task carries trust issues. Due to multiple reasons, participants sometimes provide fake or incorrect data. In order to increase data quality, this component ensures data source validity and preserves trust. A few studies focus on trust management in IoT, including the work in [109] which uses fuzzy reputation for trust management or the work in [110] which uses the hierarchical trust management model. In community sensing, trust management can be performed at two levels: at the community level and at the inter-community level. This component can work alongside the community detection component to preserve trust.

3.3 Summary

In this chapter, an architecture and its associated framework for a community-oriented smart city has been proposed. Different components of the proposed framework were detailed. Demonstration of all the components via an implementation is beyond the scope of this thesis. The primary contribution of this thesis is focused on implementing the community detection and data aggregation components which are the key functional blocks for a community-oriented smart city. Therefore, the next two chapters focus more on these two components and their implementation. The design of the architecture and its associated framework is expected to be the initial step for the community-oriented sensing research area.

Chapter 4

4 Community Detection and Analysis Using Graphs

4.1 Introduction

Since there is neither a general solution nor a unique algorithm for community identification, detecting communities in networks is an ill-defined problem in computer science [111]. Accordingly, there is no unique way to evaluate and compare the performance of different algorithms. The drawback of this ambiguity has led to considerable confusion and many misconceptions, and has also slowed down progress in this research field. However, this very vagueness gives researchers the freedom and flexibility to suggest a variety of approaches for different problems that often depend on a specific application.

This chapter proposes an approach to detect and analyze communities of individuals using graph analysis, based on assessing the similarity of patterns in sensor data over time. This demonstrates the sensing and management of data within the proposed Community-Level Sensing, Control and Management sub-layer. It further demonstrates the Community Detection and data aggregation components of Figure 13 in Chapter 3. A case study is used to demonstrate how social communities are formed based on similarity of the patterns in the sensor data. This similarity is calculated based on the sensor data collected from the smartphones of individuals. Contrary to other sensing platforms, smartphones, which are kept close to us most of the time, are part of our daily lives. Therefore, detecting communities of individuals according to data from their smart phones is more precise than other sensing platforms. As observed in the framework designed in Chapter 3, one of the main components of community analysis is community detection. As a consequence, this component forms the main focus of this chapter.

A further unique contribution of this chapter is that the proposed method transforms time series data collected from individual smartphone sensors to correlation networks by applying various similarity functions. The method then finds the communities of the corresponding correlation network by applying different community detection algorithms. In other words, the proposed method finds similar time series data and clusters them into groups; each group contains the data of the users who have the most similarity. Based on a review of the literature, this current study is the first to detect communities of individuals from the time series data collected from their smartphone sensors.

The remainder of this chapter is organized as follows. An explanation of the characteristics of communities based on reviewing the related literature is first presented followed by a discussion of the benefits of community detection in a network. The proposed method for community analysis is then described. The chapter concludes with a real case study and experimental results.

4.2 Definition of Communities

. The concept of network is almost everywhere, from social science to computer science [112]. Any network can be represented as a graph that includes edges and vertices, and where each edge connects a pair of vertices [113]. If there is a network that can be represented by graph G of which the vertices are objects or people while its edges show the correlation or similarity between the vertices, then a community is a subgraph of graph G

denoted by C, which is the number of internal edges inside the subgraph; i.e. the number of edges connecting the vertices of the subgraph C to the vertices of C is more than the number of external edges, which are the number of edges connecting subgraph C to the rest of the graph [114, 115]. Expressed simply, community is a subgraph of a graph G where the internal edges' density (internal degree) is higher than the external edges' density (external degree). Suppose that C is a subgraph of a graph G and the number of edges and the number of vertices are e, v for graph G and e_c, v_c for subgraph C. If the adjacency matrix for graph G is represented by A , then the internal and external degree for subgraph C can be shown as:

$$Internal\ degree\ of\ the\ subgraph\ C = \sum\nolimits_{i,j \in C} A_{ij} \tag{1}$$

$$External\ degree\ of\ the\ subgraph\ C = \sum_{i \in C, j \notin C} A_{ij} \tag{2}$$

where A_{ij} is an element of the adjacency matrix A, which equals 1 if the vertices i and j are connected, otherwise it equals 0.

Based on the above explanation, communities are groups of densely connected vertices with a weak connection between groups. If the vertices are considered as people and the edges are considered as similarity between people, then it can be concluded that the community is a group of people who have a common interest or similar characteristics.

Community can also be defined by measuring the probability of node connectivity. Therefore, community will be the groups of nodes that have the higher probability of being connected to each other than to nodes in other groups [111]. Figure 15 (a) shows a sample network that has four communities and, as can be clearly observed, the density of edges inside of each community is more than the density of edges outside of the community.

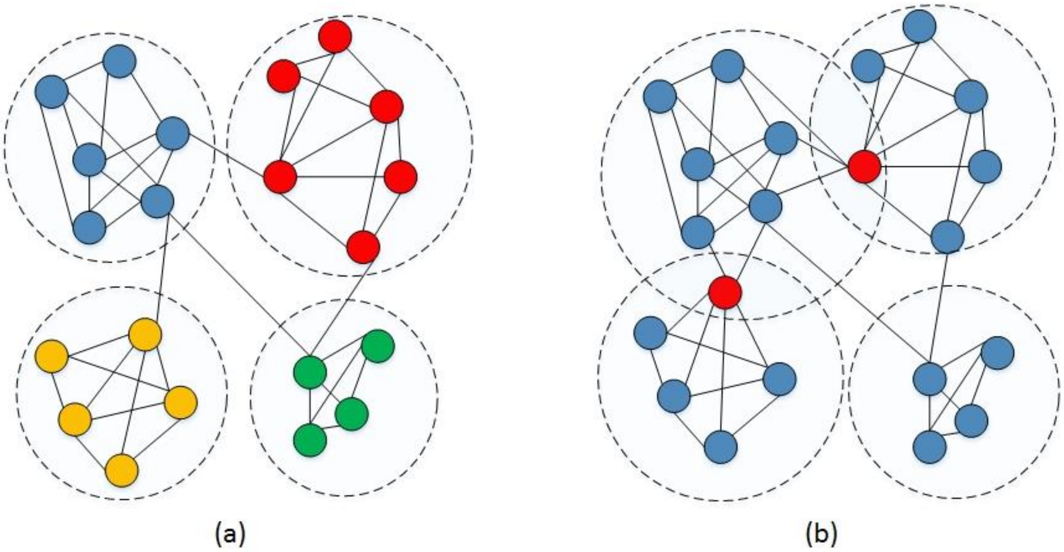


Figure 15. (a) Network with four separate communities. (b) Network with overlapping communities.

Communities are not always separated from each other and nodes can belong to multiple communities. For example, in a social network, an individual can be a member of more than one community at the same time, such as a family community, as well as a friend community. Figure 15 (b) shows the overlapping communities when one node belongs to more than one community.

4.3 Benefits of Detecting Communities

Most networks follow a community structure and their members are structured into groups that are called communities. Facebook, for instance, is a social network that connects people around the world according to their relationship. Twitter is another example where users post and interact with messages. A further well-known example for which the community concept can be used is the IoT where objects connect and interact together via the Internet. Detecting communities is one of the most important tasks when analyzing networks. Identifying communities helps us to discover groups as well as their common interests and organizational basis in the networks. Identifying communities also allows us to uncover the functionality and interactions between the network members, to predict their relations and to infer missing attributes and features. Community detection is not only limited to those networks mentioned above but has been applied to networks of many kinds, such as biological, social, human, and academic. As an example, Figure 16 [116] shows the communities in the National Collegiate Athletic Association (NCAA) football teams' network which is detected by an AGM algorithm [117]. Each color represents one NCAA team or community, the vertices represent members and the edges show the connection between them.

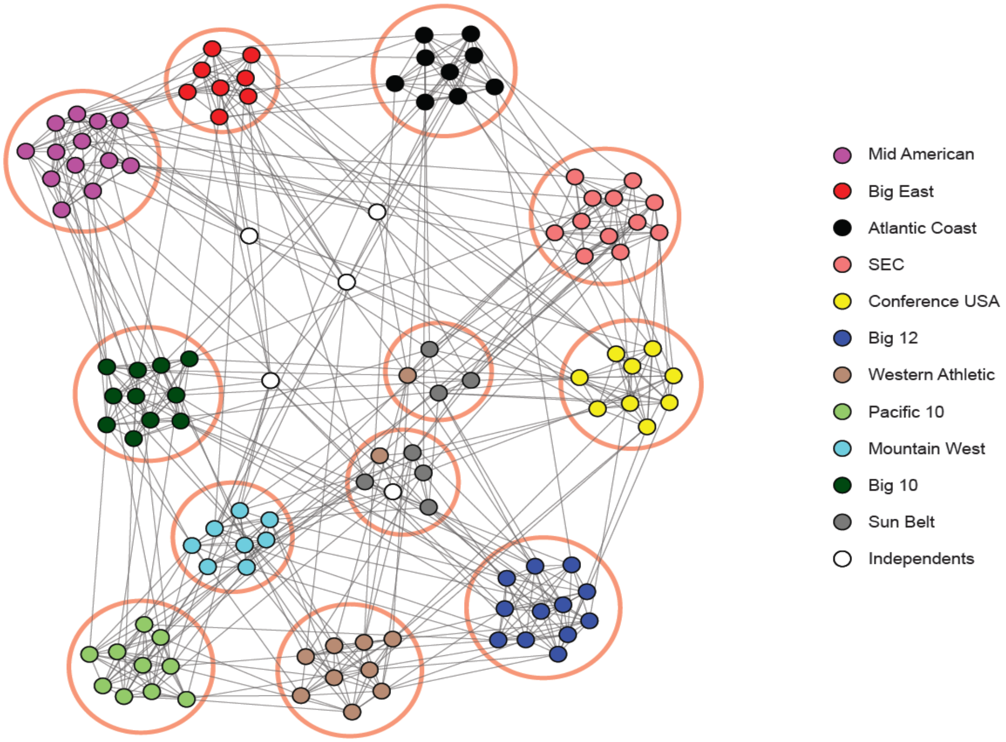


Figure 16. Communities in NCAA football team network. Figures reprinted from [116]

4.4 Proposed Method

As a large amount of the differences between people will appear as differences in their sensor data, analyzing the sensor data of various people and finding the similarity and dissimilarity between their sensor data will reveal the similarity and dissimilarity between them.

As presented in the proposed framework in Chapter 3, the main purpose of the community analysis component is to check the possibility of identifying different communities of people by leveraging their sensor data and performing a multi modal analysis on their data. In order to determine communities, a five step method is proposed. The main concept behind this method is illustrated by Figure 17 and presented in Algorithm 1. In this method, time series data is collected from users and used to construct different types of graphs known as Sensor Correlation Networks (SCNs), such as GPS or EEG correlation. Each sensor correlation network or graph is generated by inferring data from sensors originating from multiple users by using different similarity functions where each user's sensor data is represented as a vertex and graph edges indicate the level of similarity between a pair of user sensor data. In the next step, based on the application that uses the platform, the SCNs can integrate and fuse together to create a unique graph, known as a User Correlation Network (UCN), where graph vertices indicate the users and those that are the most similar are connected. For example, if there is a transportation application that needs to know the communities of users based on their movement pattern, and this application use two different sensors data, GPS and accelerometer, then both the GPS correlation network and the accelerometer correlation network can be used to generate the movement correlation network. In another example, a health correlation network can be generated by mixing EEG, ECG and heartbeat sensor correlation networks. According to the application’s needs, multiple UCNs are possible in this step. The final step is to detect the communities of users based on their UCN by applying a community detection algorithm. The proposed method provides two distinct levels of similarity: sensor similarity and user similarity. However, these two levels can be combined in one level or one of these levels can be skipped based on the applications which are used on the top layer. Although in the first level, a single SCN cannot precisely manifest the users’ similarities, it can be used to reveal the communities of the users based on a particular sensor. The steps for the proposed methods are presented below.

Algorithm 1-Community detection

```
Input:    Sensor raw data

Begin

          Data = Preprocess (Sensor Raw Data)

          SimilarityMatrices = SimilarityMeasurement (Data)

          SCNs = GraphCreation (SimilarityMatrices, k)

          UCNs = Integration(SCNs)

          Communities = CommunityDetection(UCNs)

End
```

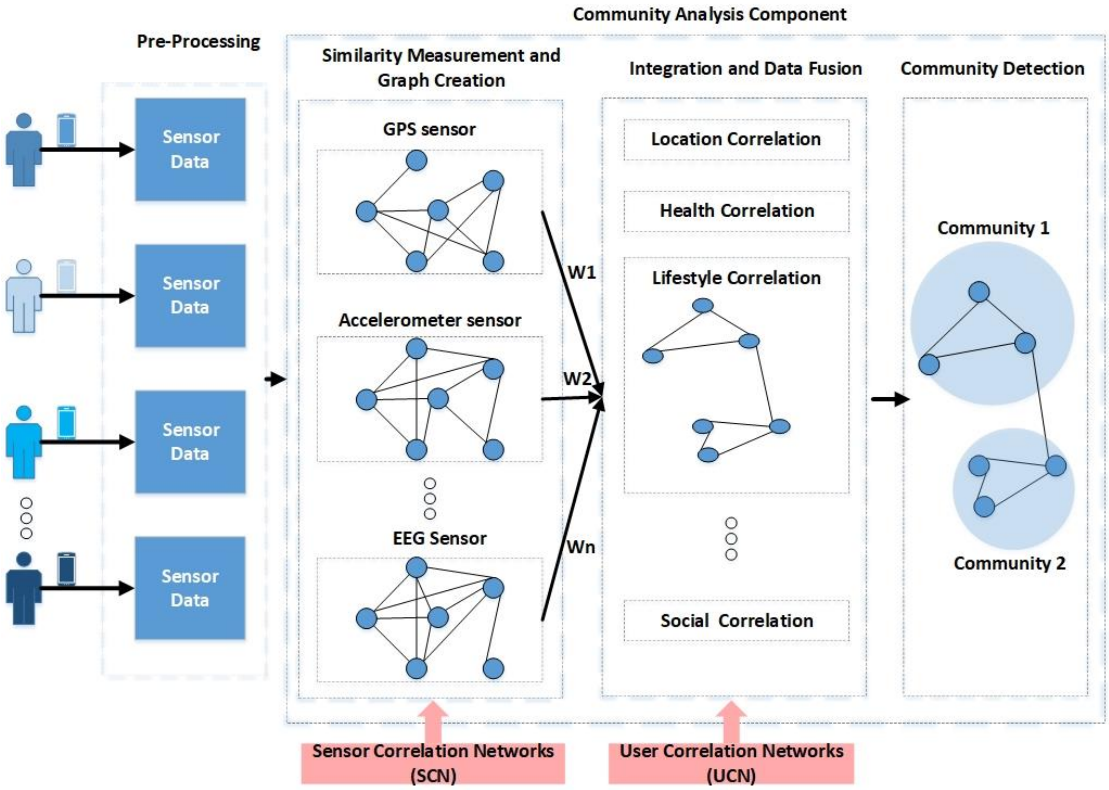


Figure 17. Community analysis component

4.4.1 Pre-processing

Pre-processing, which is the first step in the presented method, transforms the sensor raw data into a format that is more effectively processed and then prepares this data for the subsequent steps. Pre-processing is a very important step because it directly affects the community detection result. This step includes noise filtering, data normalizing and vectorization.

4.4.2 Similarity Measurement

The many differences between users will by reflected by the differences in their sensor data. Therefore, measuring the similarity or correlation between their sensor data can reveal their similarity.

The second step in the proposed method is to measure the similarity or correlation between users based on their sensor data and then create a similarity matrix S for each sensor, where each element (s_{ij}) in the matrix S is the similarity value between user U_i sensor data and user U_j sensor data. This will result in the same number of similarity matrices as the number of sensors. For instance, if a user has data for GPS, EEG and ECG sensors, three different similarity matrices will then be created: one for each of the GPS and EEG sensors and the third for the ECG sensor. Users' social data, such as Twitter, can also be considered as a virtual sensor and can also form a similarity matrix. Assuming there is a similarity function to measure the similarity between the sensor data of two users, then:

$$s_{ij} = Similarity\ (U_i, U_j)^{Sensor_z} \tag{3}$$

where s_{ij} is an element of similarity matrix S and z specifies the sensor type. The similarity function would be different from one sensor to another.

Because users' sensor data are time-series so the similarity between two users can be calculated by finding the similarity between two time-series.

In the proposed method, every SCN is created by using its corresponding similarity matrix, which is created in the current step. Nodes in the SCN represent the users' sensor data and edge-weights represent how the users' sensor data are similar. To express it in a simple way, edge-weights are created based on similarity matrix S. The proposed framework is designed to measure the similarity of multiple sensors in order to capture the various levels of dependency among users. Depending on the number of sensors with which each user is equipped, there can be a variety of SCNs.

Measuring the similarity between users' raw sensor data is a complicated task because most of the sensors create high dimensional data and computing the level of similarity between them is complex. Thus, instead of measuring the similarity between raw sensor data, it is possible to measure the similarity between the features extracted from the raw data. To this purpose, the features for each user sensor data can be used to construct a histogram to measure the similarity between them. Based on the type and number of features, this histogram can be either one or two dimensional or, in some cases, even more than two dimensional. For instance, for a GPS sensor that considers the features as latitude, longitude and time, then the histogram can have three dimensions. Thus, if considering T_i and T_j are the histograms of user $U_i$and user U_j then:

$$s_{ij} = Similarity\ (T_i, T_j)^{Sensor_z} \tag{4}$$

One of the well-known similarity functions is Dynamic Time Wrapping (DTW) [118], which aligns two time series (histograms in this case) by employing the shortest wrapping path in their distance matrix and calculating the similarity between them.

$$s_{ij} = DTW(T_i, T_j) \tag{5}$$

Given two time series Q and C of length n (K - dimensional), aligning these series using DTW involves first having to build a $n - by - n$ matrix where the d_{ij} (i^{th}, j^{th}) element of the matrix corresponds to the Euclidean distance of point i in Q and point j in C.

$$d_{ij}(Q,C) = \sqrt{\sum_{k=1}^{k} (q_{ki} - c_{kj}) * (q_{ki} - c_{kj})} \quad (6)$$

The second step is to identify a path through the matrix that minimizes the total cumulative distance between them or, in other words, that minimizes the wrapping cost. The DTW is computed based on the following equation:

$$DTW(Q,C) = \min \sqrt{\sum_{z=1}^{Z} P_z} \quad (7)$$

where P_z is the z^{th} element of the wrapping path P, a set of distance matrix elements that indicates a mapping between Q and C.

4.4.3 Graph Creation

By having a similarity matrix, an SCN for each sensor can be created. Each node in the graph represents a user and the edges' weights between nodes represent the similarity among users. Thus, if two users are similar, there will be an edge between them; if they are not similar, there will be no edge between their corresponding nodes. The two most popular graph creation algorithms are K nearest neighbor (K_{NN}) and $\in$ nearest neighbor $(\in_{NN})$ [119]. The first algorithm connects each node to the K most similar nodes and the second algorithm considers a threshold $\in$ and connects each node to another if the similarity between them is more than the defined threshold.

4.4.4 Integration

After all the SCNs are constructed, they are fused together to create UCNs. This integration will depend on the applications that are used on the top layer. Each application will have a different weight for each sensor. The similarity matrix (U) of each UCN is created by the following equation:

$$u_{ij} = \sum_{z=1}^{n} W_z * S_{ij}^{z} \qquad (8)$$

where U_{ij} is an element of the UCN similarity matrix(U), indicates the similarity between user i and user j, W_z is the given weight for the z^{th} SCN, S_{ij}^{z} is an element of the z^{th} SCN indicates the similarity between user i and user j sensor data and z specifies the sensor's type that is used to create the UCN.

After building the similarity matrix for each UCN, the same graph creation methods that were used in the previous step can be used to build the UCNs. The number of SCNs involved in creating a single UCN depends on the type of UCN. For instance, a Lifestyle UCN can be created by using GPS, EEG and ECG SCNs.

4.4.5 Community Detection

This is the final step for community analysis. In this step, the community detection algorithms are applied to the UCN in order to identify some different communities within it. Each community or cluster is an indication of the group of highly connected users who share a similarity in different aspects, depending on the type of UCN. Many community detection algorithms are available [114, 120-123]; the best result will depend on the selection of the algorithm. The most common algorithms are Louvain [120], LeMartelot

[121], Newman's Greedy Algorithm (NGA) [122], and Danon. In this section, a brief review of each algorithm is given.

The Louvain Algorithm [120]: Louvain is a heuristic method based on modularity optimization. In clustering optimization, the goal is to maximize the function (objective function) which indicates the quality of a clustering, over all possible clustering space. In the Louvain algorithm, the quality of clustering is measured by modularity of the partitions so the goal is to maximize the modularity. The modularity can be defined as a scalar value between -1 and 1, which measures a density of edges inside clusters/communities to the density of edges outside clusters/communities. Therefore, the algorithm tries to find the optimized clusters by maximizing the density inside of communities and minimizing the density between communities. The modularity function in Louvain algorithms is defined as follows:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \qquad (9)$$

where $\sum_{in}$ is the sum of the weights of the links inside community C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in community C, k_i is the sum of the weights of the links incident to node i, $\sum_{i,in}$ is the sum of the weights of the links from i to nodes in community C and m is the sum of the weights of all the links in the network.

A Louvain algorithm is performed in two steps, repeated iteratively. In the first step, each node (user in this thesis) is assigned to a separate community and then modularity calculated if the nodes are removed from their own communities and assigned to their neighbour's communities. In the second step, the Louvain algorithm assigns the node to the community which maximizes the modularity. This step is repeated until no change occurs in modularity value for all nodes.

LeMartelot [121]: LeMartelot is another community detection algorithm based on greedy optimization and is similar to the Louvain algorithm. However, LeMartelot uses the stability function along with the Markov process to measure the quality of communities instead of modularity.

Newman's Greedy Algorithm [122] : NGA is another community detection method based on modularity optimization which identifies communities in a large scale network. NGA algorithms work similar to Louvain algorithms but with a different modularity function. The modularity value is defined as follows [120] :

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i c_j) \tag{10}$$

where A_{ij} is the similarity between user i and user $j, k_i$is the sum of similarity value of the nodes attached to node i, C_i is the community to which that user or node i is attached, m is the sum of all similarity values in the graph and δ is a simple delta function.

4.5 Use Case Scenario and Experimental Results

This section provides an evaluation of the effectiveness of the proposed method and presents the experimental results from the proposed community detection method. The experiment shows that collecting the sensor data of individuals can effectively identify their belonging communities and determine the similarities among them.

4.5.1 Experiment Settings and Methodology

4.5.1.1 Objective

The objective of this experiment is to establish the presence of communities of users according to their mobility patterns and frequency of visiting locations, as well as to verify the accuracy and performance of the selected methodology.

4.5.1.2 Datasets

This experiment uses location data which was collected over the course of a month (November to December 2016) from 14 users who live in the Greater Toronto Area (GTA) in Canada. The location data was collected from the GPS sensors of their smartphones by using Google location history software, and transferring it to the cloud via a 3G/4G network connection. It was then stored in an SQL server database on the cloud. The users are undergraduate and graduate university students as well as non-students. The collected data contains times, GPS coordinates (latitude, longitude, altitude). The mobility pattern is considered as a time series of GPS coordinates. Figure 18 to Figure 32 show the users' movement patterns in the total periods of data collection. In Figure 18 to Figure 31, a line was drawn between each consecutive data point and overlaid on a Google map, but with no timestamp. Figure 32 includes the timestamp as one of the figure dimensions.

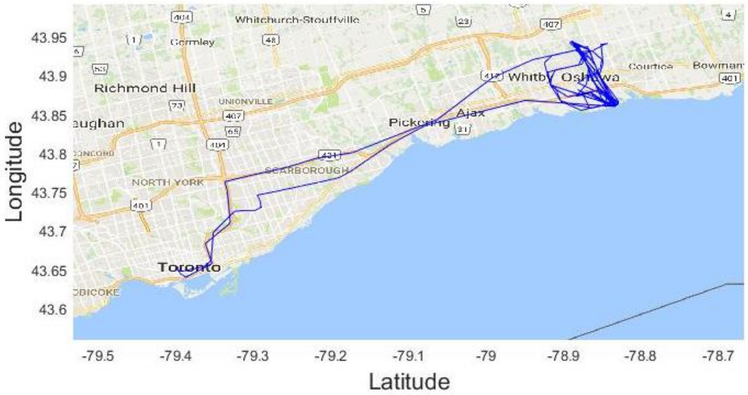


Figure 18. User 1 movement patterns

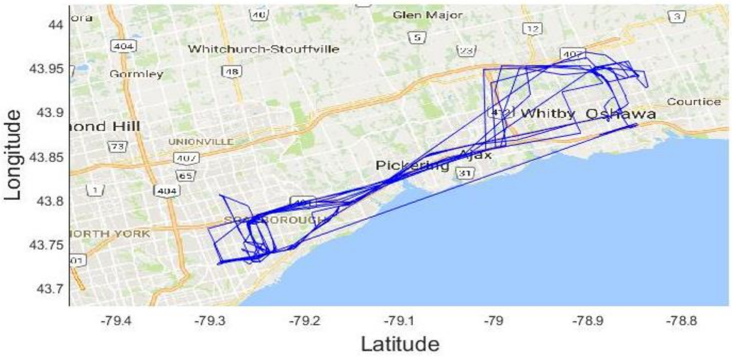


Figure 19. User 2 movement patterns

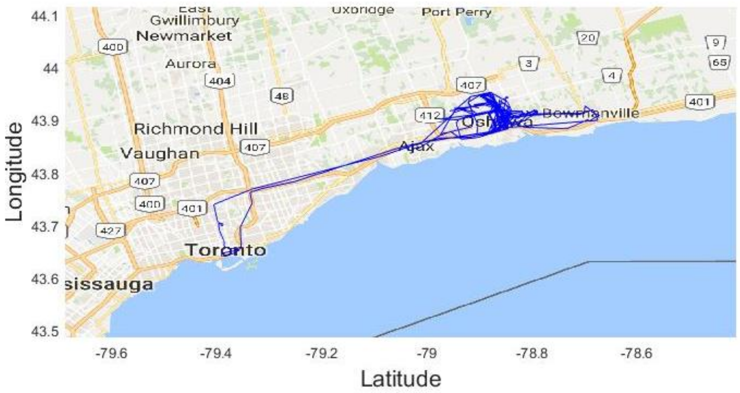


Figure 20. User 3 movement patterns

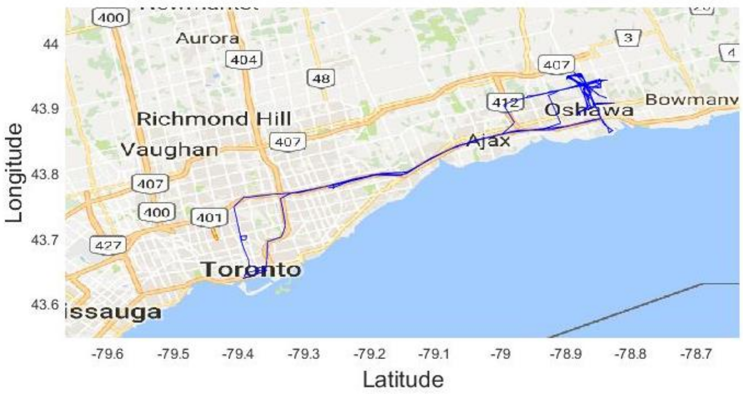


Figure 21. User 4 movement patterns

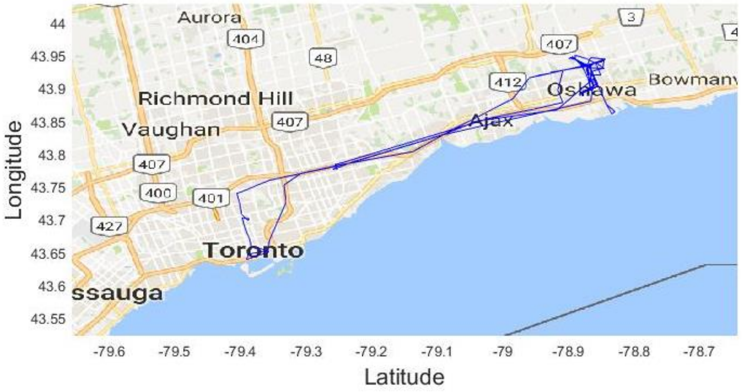


Figure 22. User 5 movement patterns

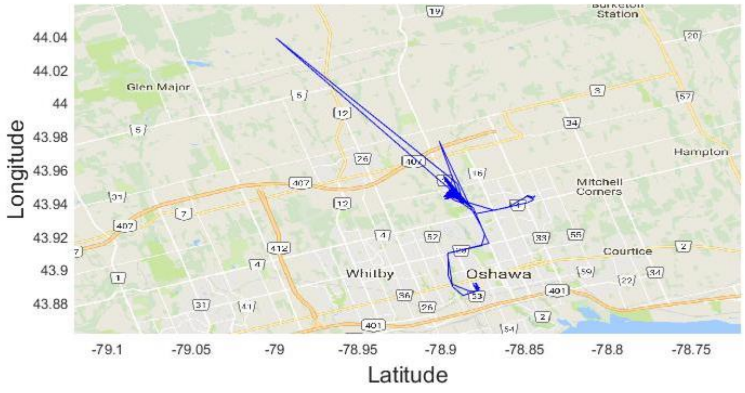


Figure 23. User 6 movement patterns

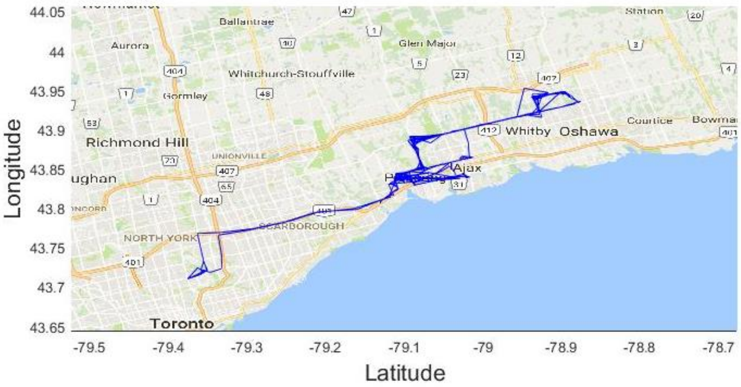


Figure 24. User 7 movement patterns

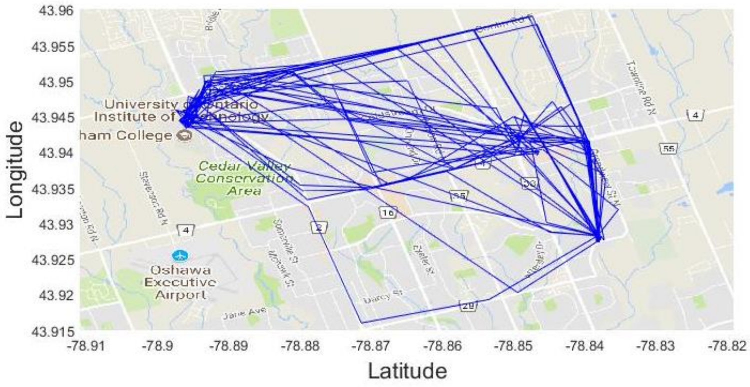


Figure 25. User 8 movement patterns

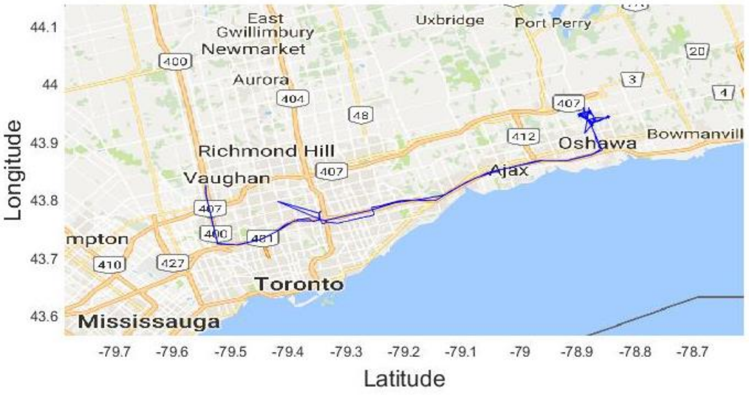


Figure 26. User 9 movement patterns

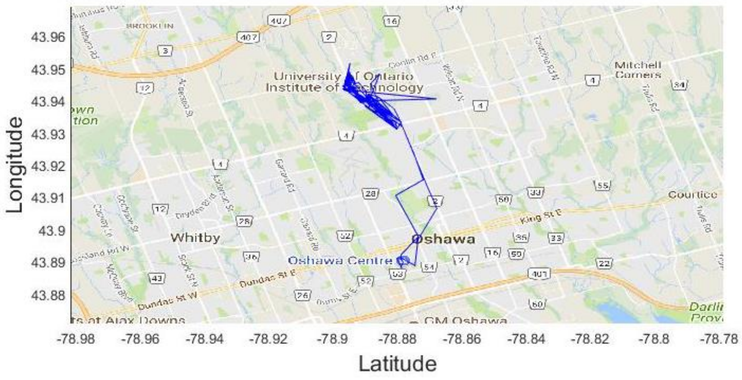


Figure 27. User 10 movement patterns

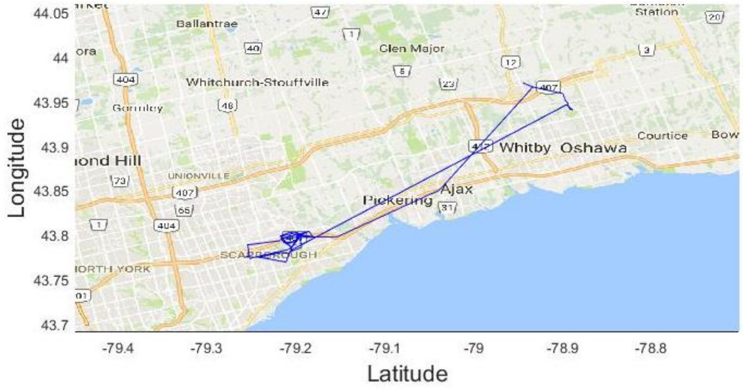


Figure 28. User 11 movement patterns

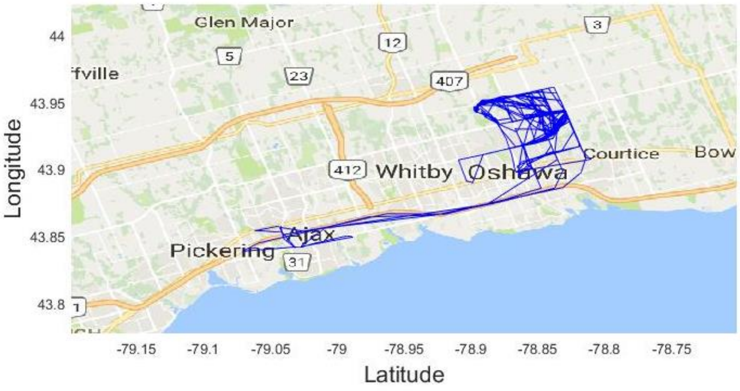


Figure 29. User 12 movement patterns

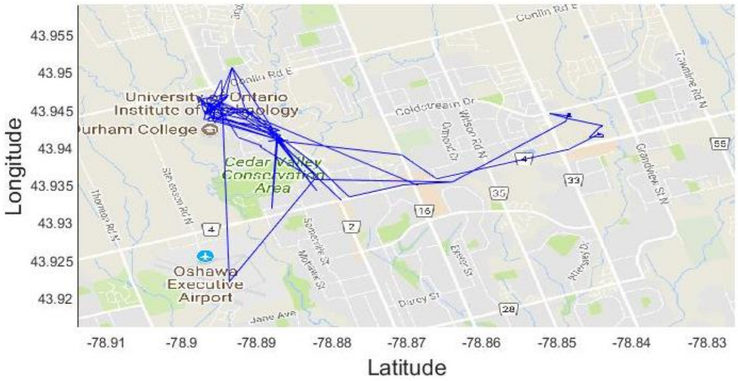


Figure 30. User 13 movement patterns

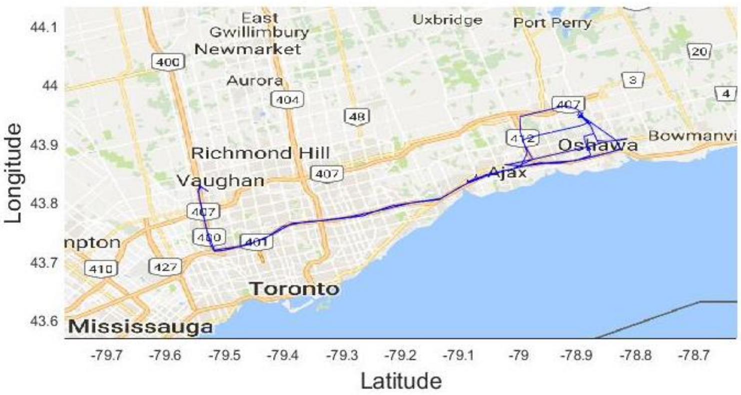
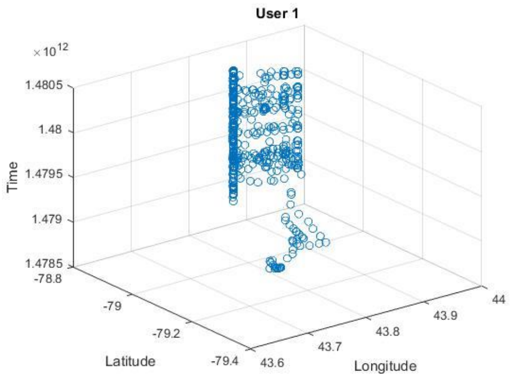
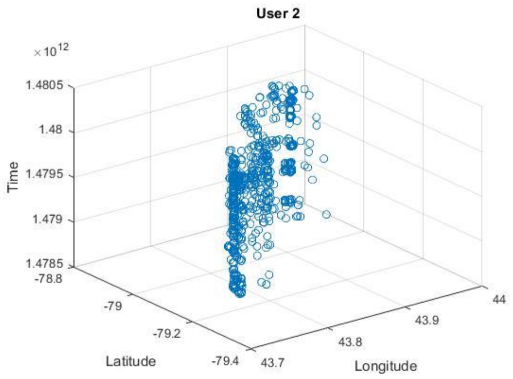
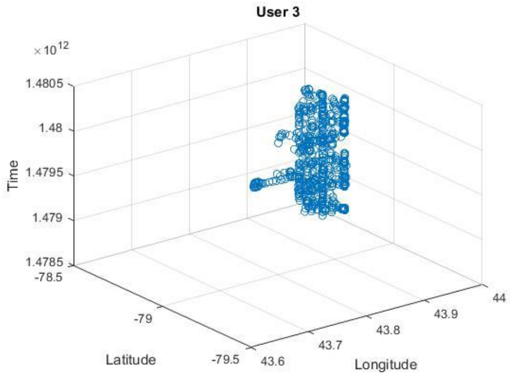
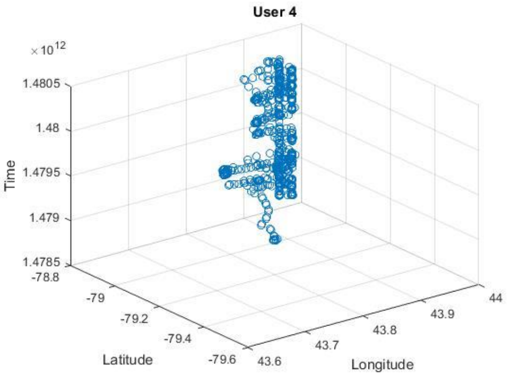


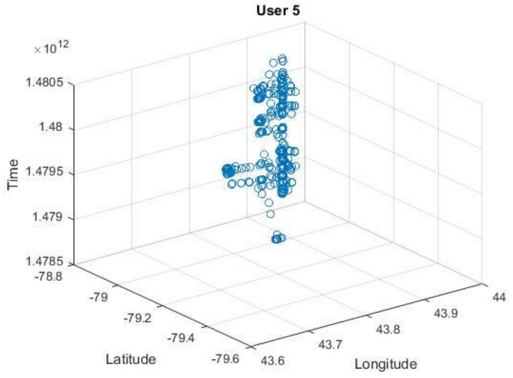
Figure 31. User 14 movement patterns

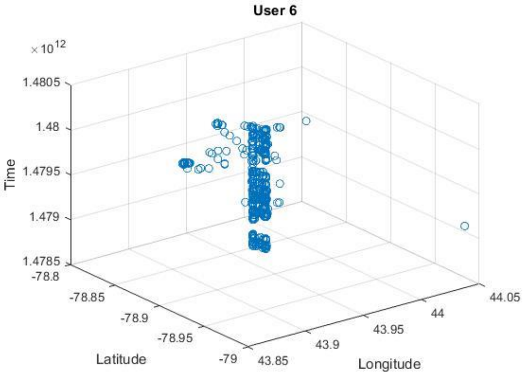

User 1

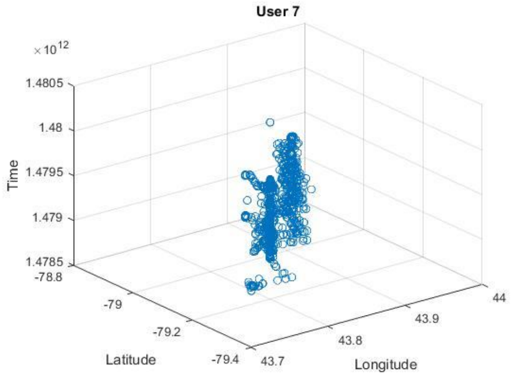

User 2

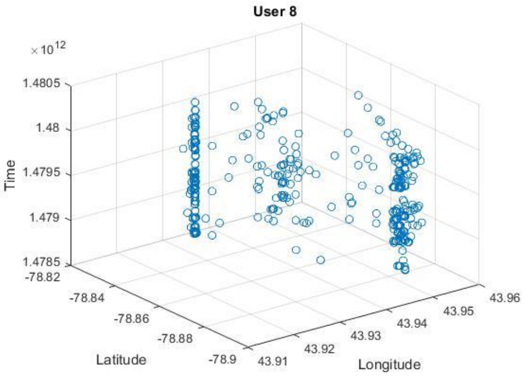

User 3


User 4


User 5


User 6


User 7


User 8

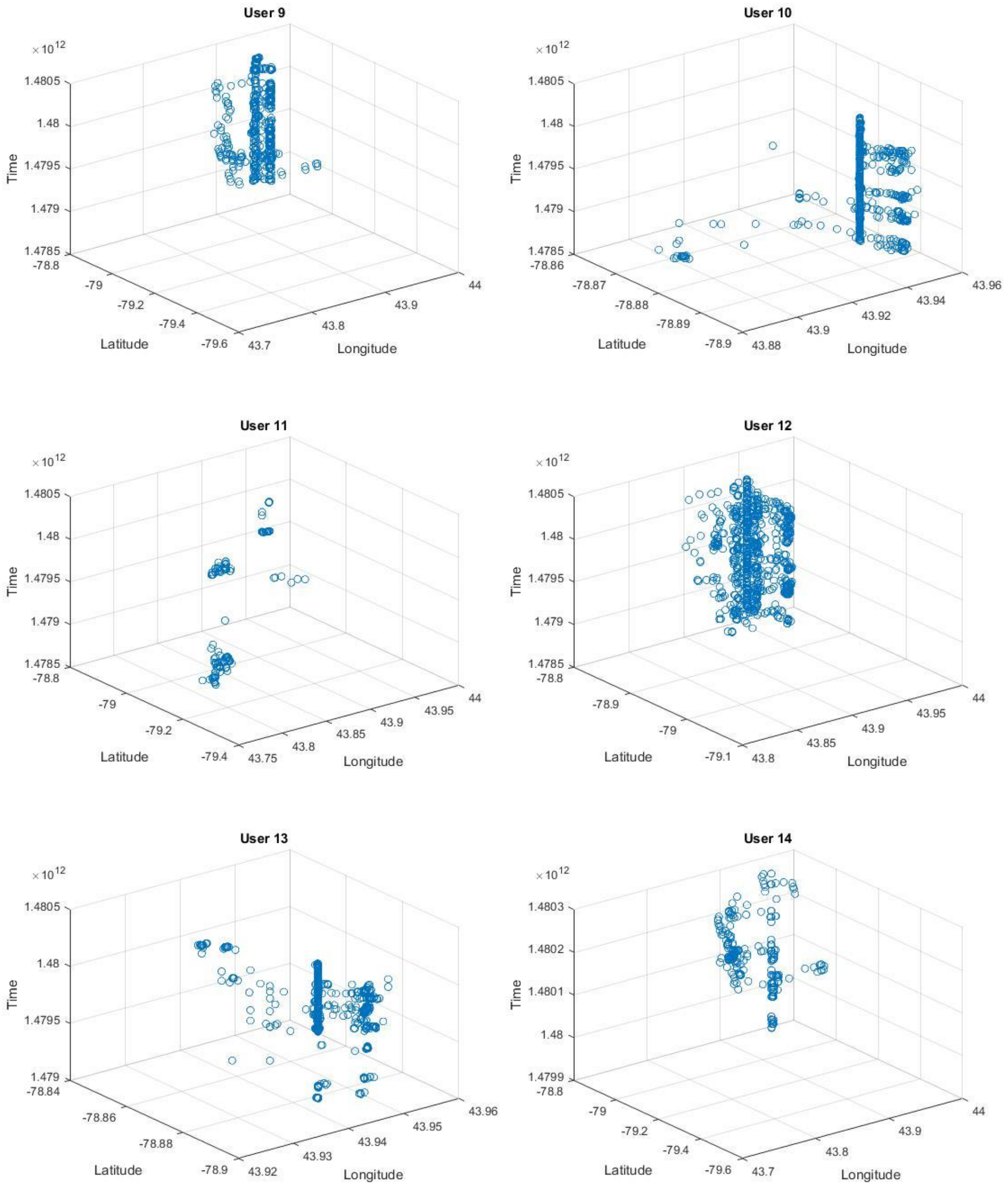
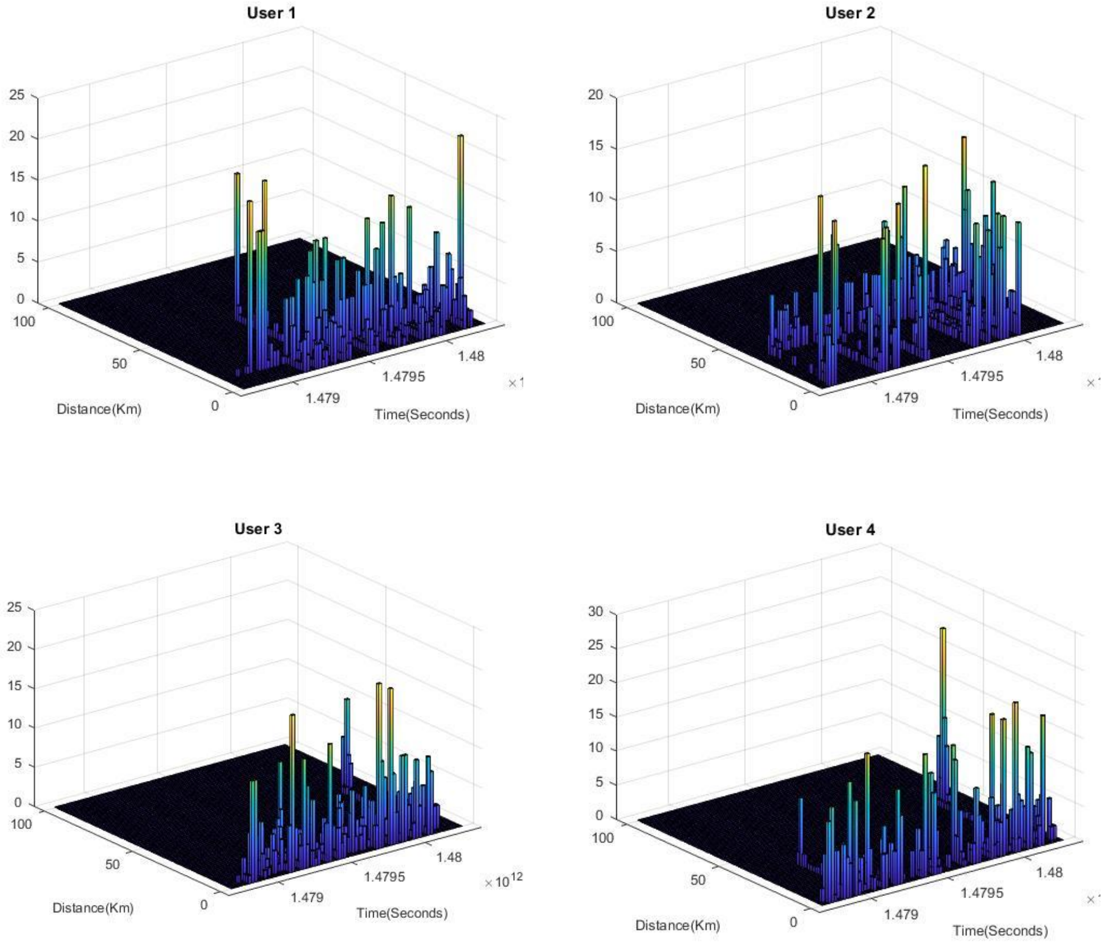
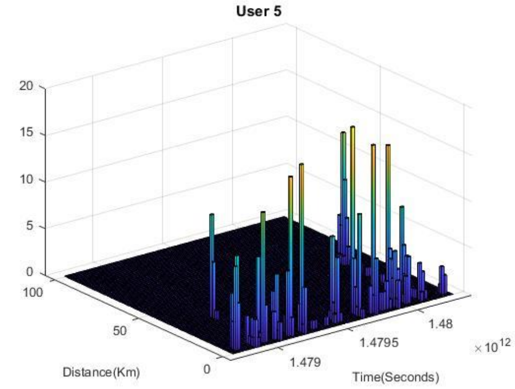


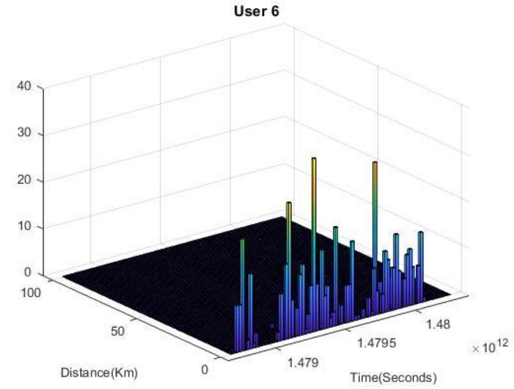
Figure 32. User movement patterns with time (user 1 to user 14)

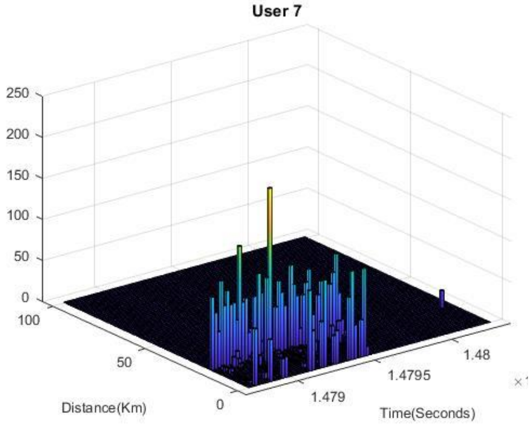
4.5.1.3 Data Processing

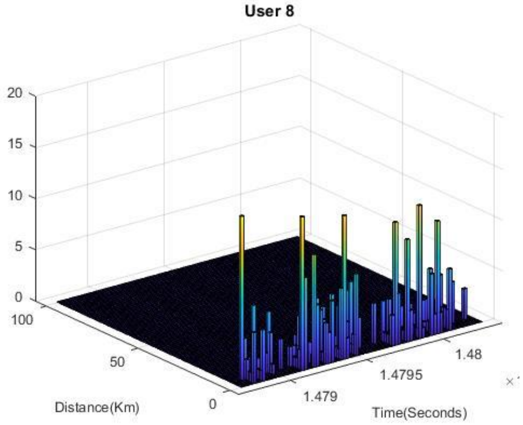
For each user, the data (longitude, latitude and timestamp) is tessellated to three-dimensional equal-sized tiles. Each tile is considered as a bin and a 3D histogram is created for each user. Figure 33 shows the users' histograms. For simplicity, GPS coordinates are converted to one feature, which is the distance from a known location (the campus of the University of Ontario Institute of Technology, Oshawa). The values of the histogram vector show the distribution of data for each user.

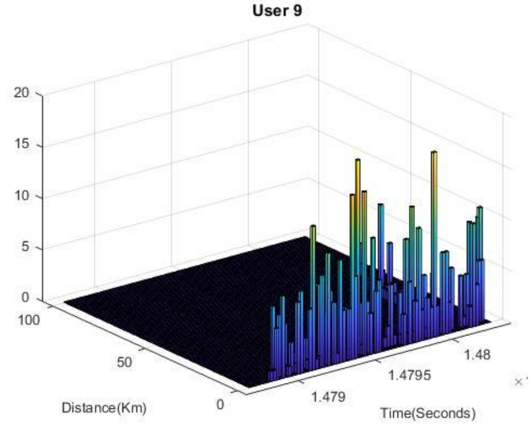


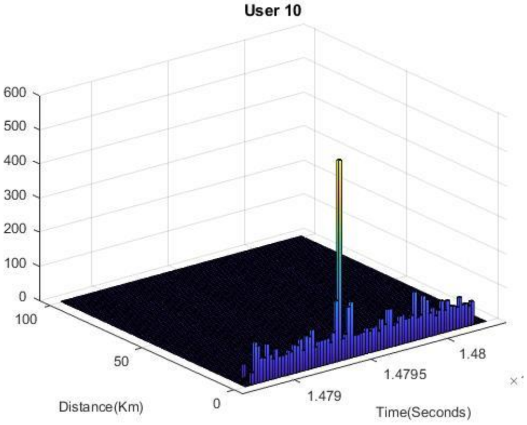

User 5
20
15
10
5
0
100
50
0
Distance(Km)
1.479
1.4795
1.48
×10^12
Time(Seconds)


User 6
40
30
20
10
0
100
50
0
Distance(Km)
1.479
1.4795
1.48
×10^12
Time(Seconds)


User 7
250
200
150
100
50
0
100
50
0
Distance(Km)
1.479
1.4795
1.48
Time(Seconds)


User 8
20
15
10
5
0
100
50
0
Distance(Km)
1.479
1.4795
1.48
Time(Seconds)


User 9
20
15
10
5
0
100
50
0
Distance(Km)
1.479
1.4795
1.48
Time(Seconds)


User 10
600
500
400
300
200
100
0
100
50
0
Distance(Km)
1.479
1.4795
1.48
Time(Seconds)

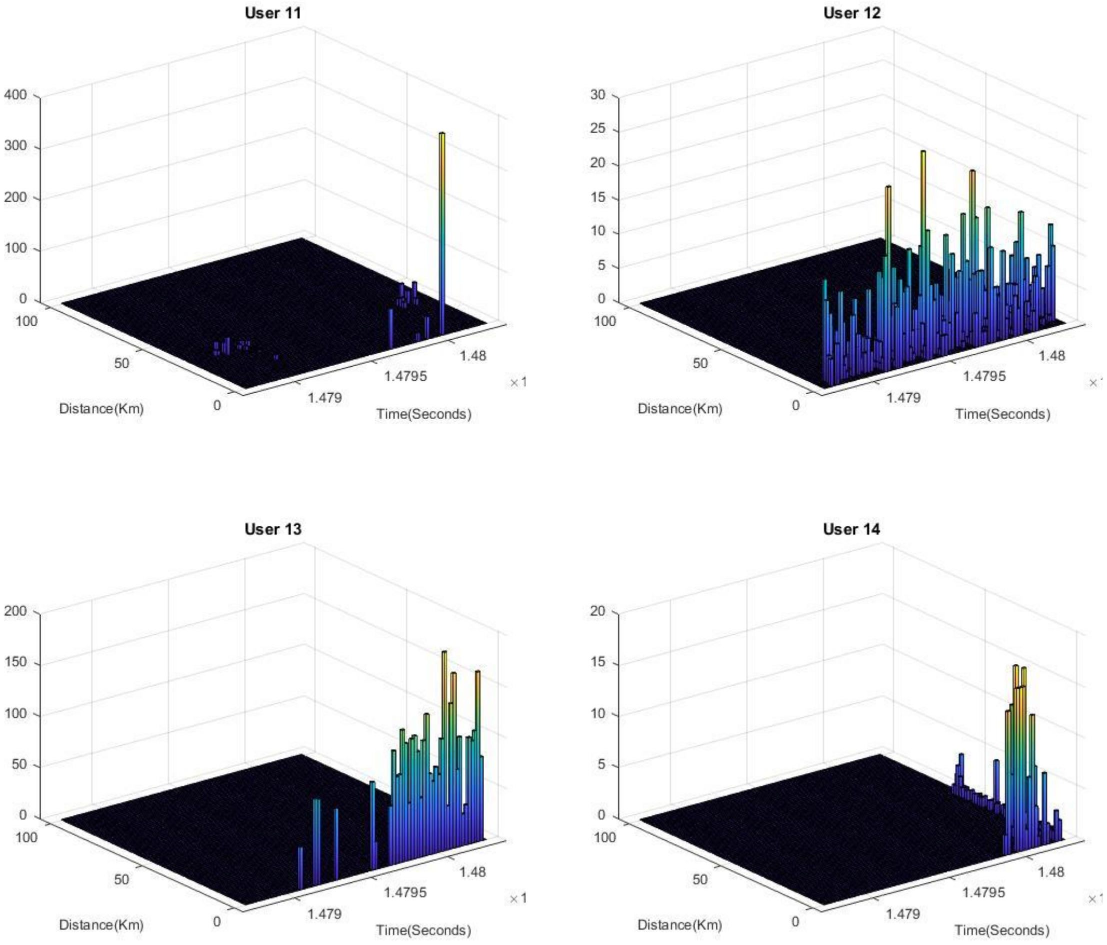
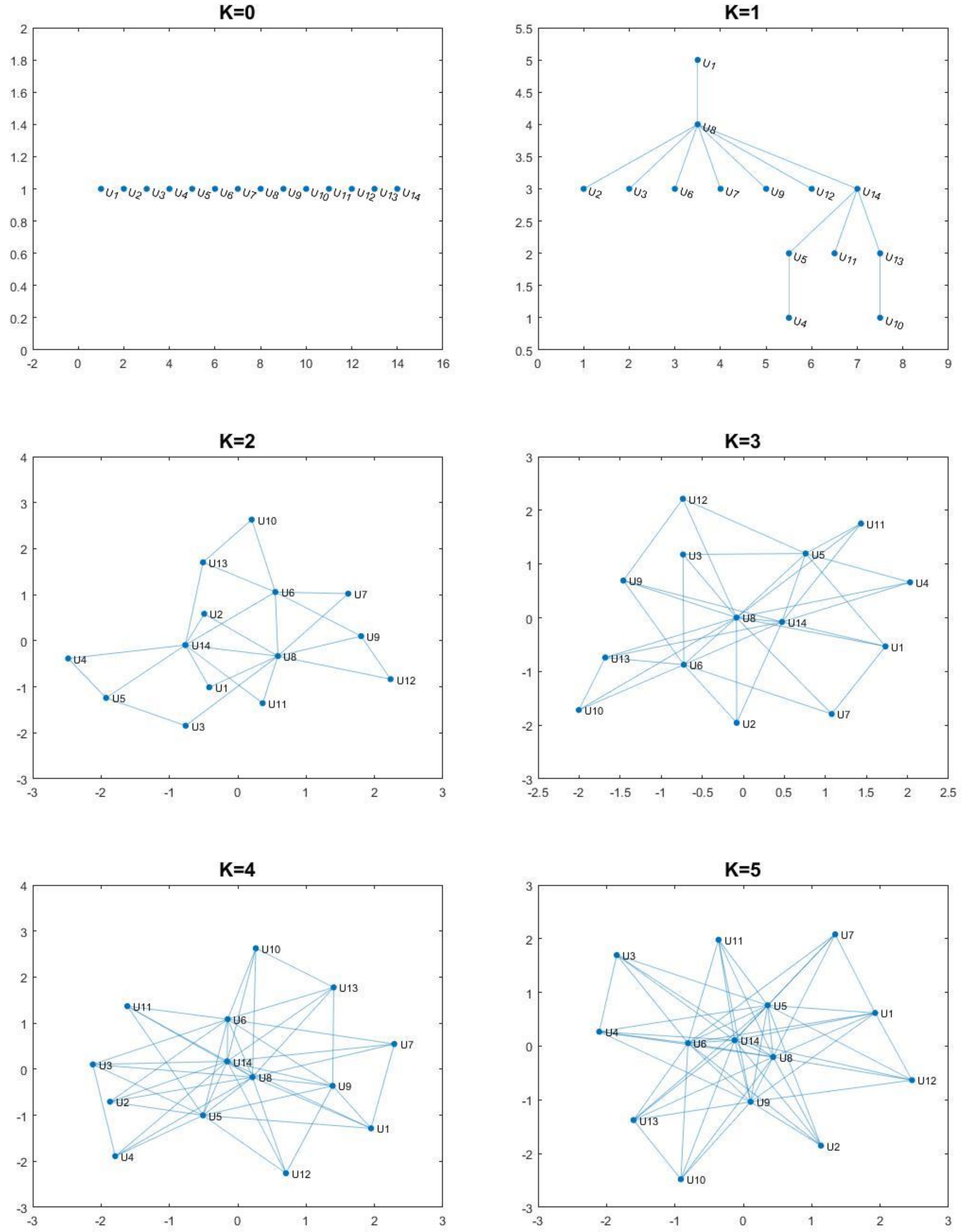
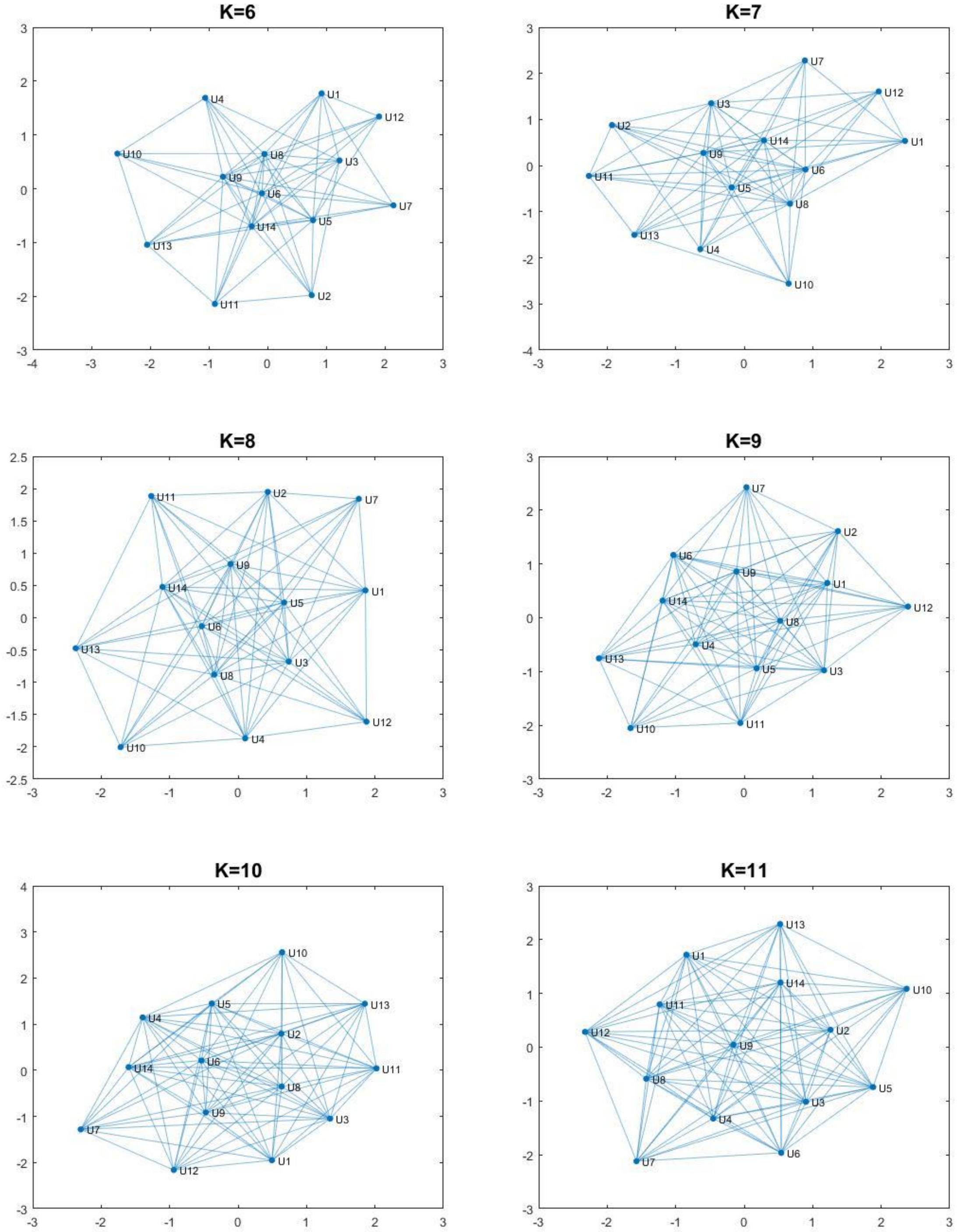


Figure 33. User location histogram (user 1 to user 14)

The subsequent step involved identifying a similarity between users and constructing a similarity graph based on similarity values. For this purpose, DTW was used as a distance metric and a similarity matrix created by calculating the similarity between users' histograms. After building the similarity matrix for all users, a well-known graph creation algorithm, K_{NN}, was employed in order to build a similarity graph. In K_{NN}, each node is connected to its K most similar nodes. The value of K has a direct effect on the community detection algorithm. K can be considered as an average degree of the graph or network. Figure 34 shows the different similarity graphs by changing the value of parameter

K(average degree). As can be observed, when K is set to “0”, the graph is completely disconnected and every node is placed in a different community. When K is set to “13”, the result is a fully connected graph where every node belongs to one single community.





K=6

K=7

K=8

K=9

K=10

K=11

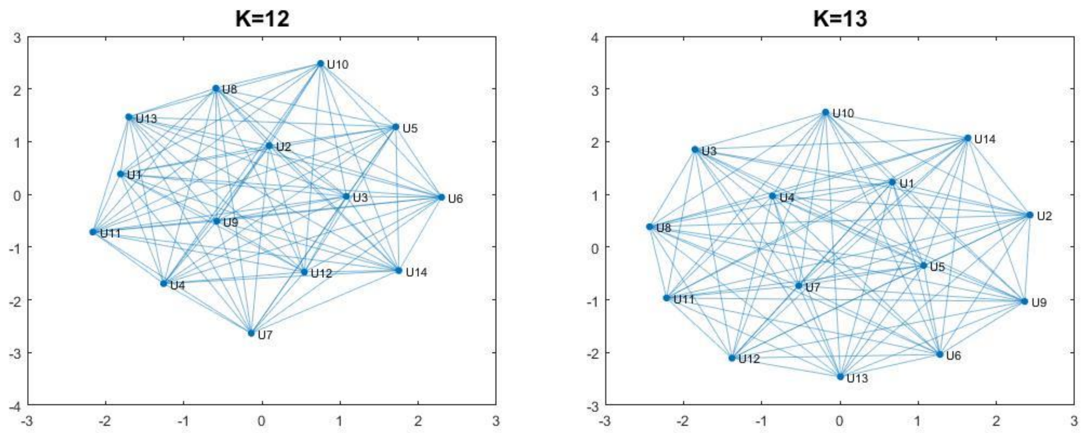


Figure 34. Similarity graphs constructed with K-NN algorithm. K changes from 1 (totally disconnected graph) to 14 (fully connected graph)

The next step is to apply a clustering algorithm to the previous step graphs and determine users' communities. In order to identify the communities, four different algorithms are employed: Louvain [120], LeMartelot [121], NGA [122], and Danon.

4.5.2 Experimental Evaluation

In order to validate the proposed method, it is necessary to check how accurately the method identifies communities. One approach is to check the precision by using benchmark networks whose community structures are known and determine how the applied method recovers communities in these networks. Another approach involves measuring the partition similarity, i.e., the similarity of the detected and actual communities. As there is no unique definition for community in the literature, identifying a benchmark network is difficult and may be arbitrary. As a result, this experiment uses partition similarity for validation.

The objective of this experiment is to evaluate the performance of the proposed method by checking the different combinations of algorithms used in each step. To compare the results with planted communities, three different partition similarity metrics are used: the Rand Index (RI), the Jaccard Index (JI) and the Normalized Mutual Index (NMI). The RI, which is a well-known pair matching similarity index, is the ratio of the correctly classified vertices in communities to the total number of vertices.

$$RI = \frac{A + B}{n(n-1)/2} \tag{11}$$

where A is the number of pairs of vertices truly classified into the same cluster, B is the number of pairs of vertices truly classified in the different communities and n is the total number of vertices. The RI for this experiment is computed by comparing its results to the correct community labels provided manually before the experiment. The RI has a value between [0, 1] where "0" indicates independent partitions and "1" indicates identical partitions.

The JI [124]is another partition similarity index in which the number of pairs of vertices are truly classified in the same cluster (A) by the number of pairs in the same community in one partition and different communities in the other partition (C), plus the number of pairs of vertices which are in the same community in both partitions (A):

$$JI = \frac{A}{A + C} \tag{12}$$

The NMI [125], another well-known partition similarity metric, is used for evaluating community detection results. The NMI is the normalized version of the Mutual Information (MI) used in information theory to scale results between [0,1] where "0" indicates no mutual information or independent partitions and "1" indicates a perfect correlation or identical partitions. NMI is computed as shown below:

$$NMI(x, y) = \frac{2MI(X, Y)}{H(X) + H(Y)} \tag{13}$$

where x and y are the two partitions whose similarity is measured, X and Y are two random variables with the joint probability distribution of $P(X, Y) = n_{xy}/n$, MI is the mutual information which is calculated according to the Shannon entropy [125] of the variable X and conditional entropy of X given Y and is equal to $MI(X, Y) = H(X) - H(X|Y)$, $H(X)$ and $H(Y)$ are the Shannon entropy of variables X and Y.

In order to calculate the Shannon entropy, the following equations [125] are used:

$$H(X) = -\sum\nolimits_{x} P(X) \log P(X) \tag{14}$$

where $P(X)$ is the marginal probability distribution function of X.

$$H(X|Y) = -\sum\nolimits_{x,y} P(X, Y) \log P(X|Y) \tag{15}$$

where $P(X, Y)$ is the joint probability distribution function of X and Y.

4.5.2.1 Effect of graph creation on applied method

In the first scenario, the effect of the graph construction on the community detection method is evaluated. The effect on the community detection process of the parameter K in the $K - NN$ algorithms is determined. An additional step changes the K from 1 to 13 and the partition similarity is measured with three different metrics: NMI, RI and JI. Table 6 indicates the first scenario set-up parameters.

Table 6. Set-up parameters for the first scenario

Description	Value
Similarity Function	DTW
Graph Creation Algorithm	K-NN
Average Degree of the similarity graph (K)	1-13
Community Detection algorithm	Louvain, LeMartelot, NGA, Danon
Number of Users	14

Figure 35 to Figure 38 show the effect of K on partition similarity when the Louvain, NGA, LeMartelot and Danon algorithms are used. As can be observed in all cases, when K is large or small, the partition similarity indices are low and the community detection method cannot effectively discover communities but the method has its best partition similarity when K is in the middle. This is because, when K is equal to 1, there are fewer links between the nodes and the community detection algorithm detects a higher number of communities. When K is large, the majority of vertices are connected and the community detection method finds a smaller number of communities. All indices show that setting the average degree to 6 achieves the best partition similarity.

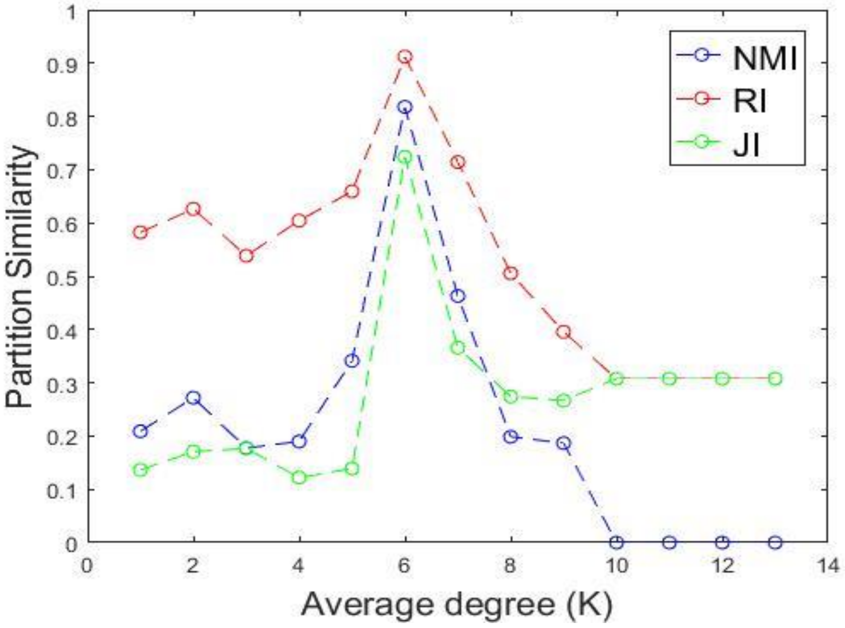


Figure 35. Effect of K on partition similarity when Louvain algorithm is used

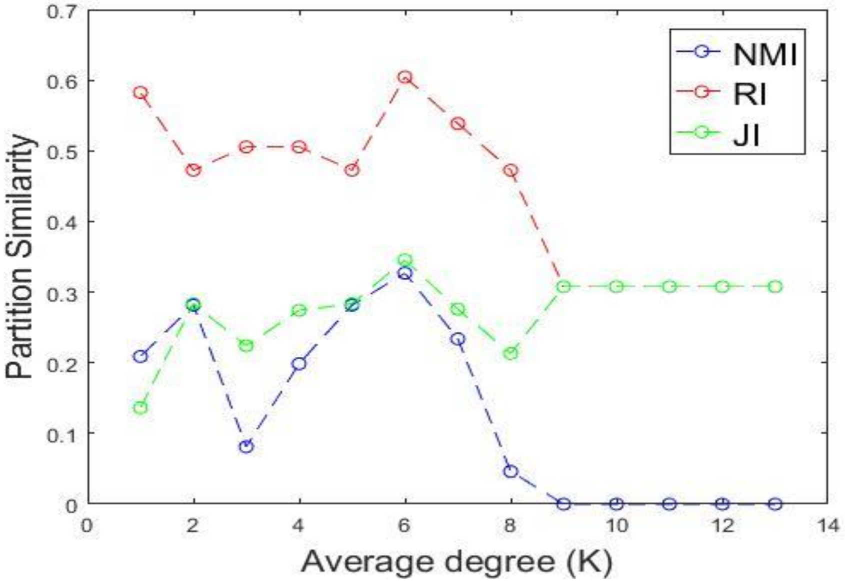


Figure 36. Effect of K on partition similarity when NGA algorithm is used

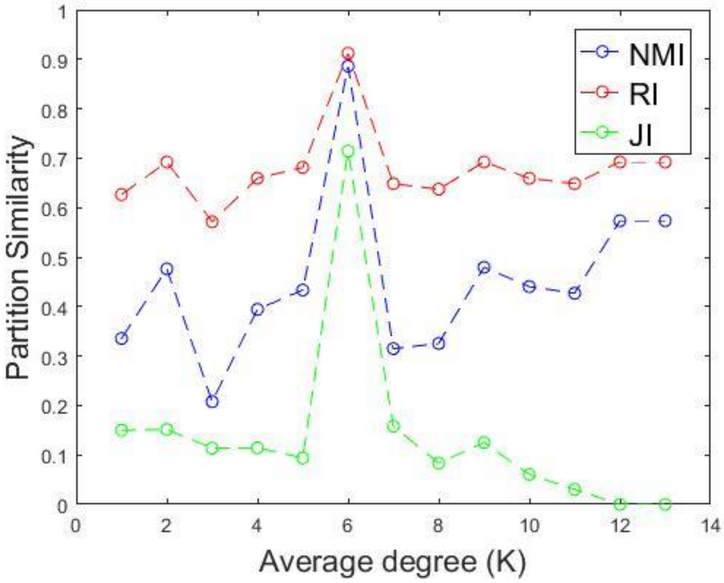


Figure 37. Effect of K on partition similarity when LeMartelot algorithm is used

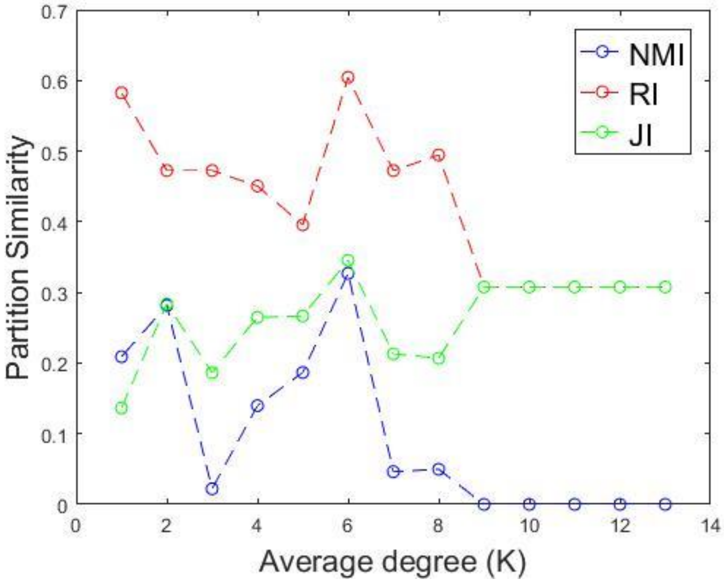


Figure 38. Effect of K on partition similarity when Danon algorithm is used

Figure 39 shows the effect of K (average degree of vertices) on the number of communities. Figure 40-43 show the different communities that are identified when the K value is equal to 1, 6 and 13 when using different community detection algorithms.

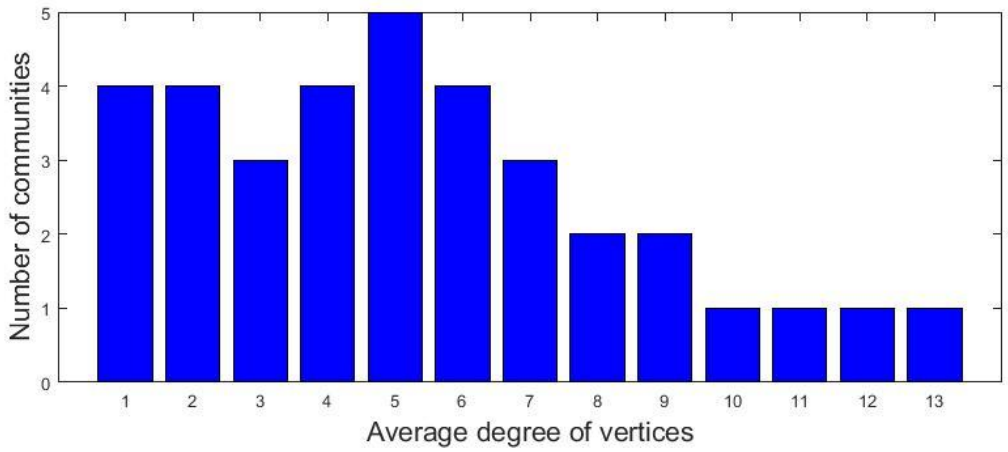


Figure 39. Effect of K (average degree) on the number of communities

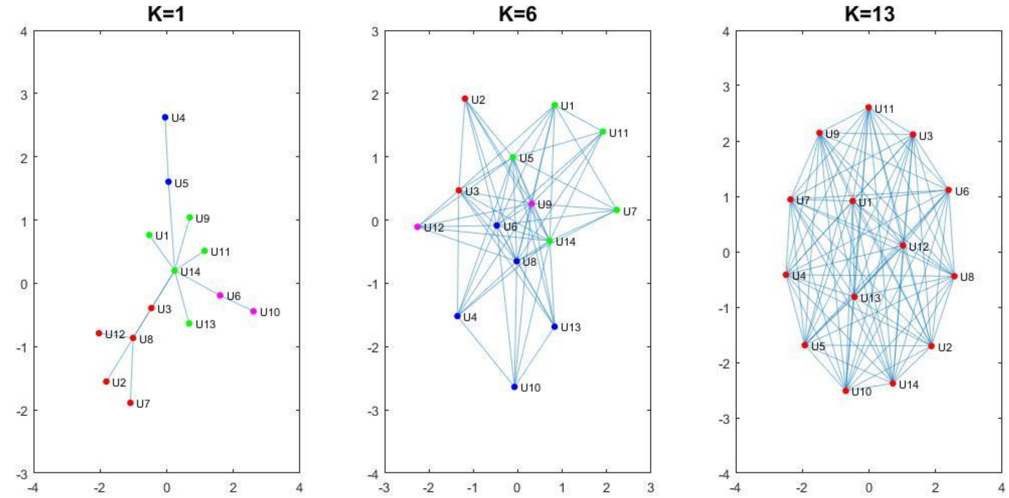


Figure 40. Communities detected when Louvain algorithm is used. Node color represents the communities found in each graph and K represents average degree

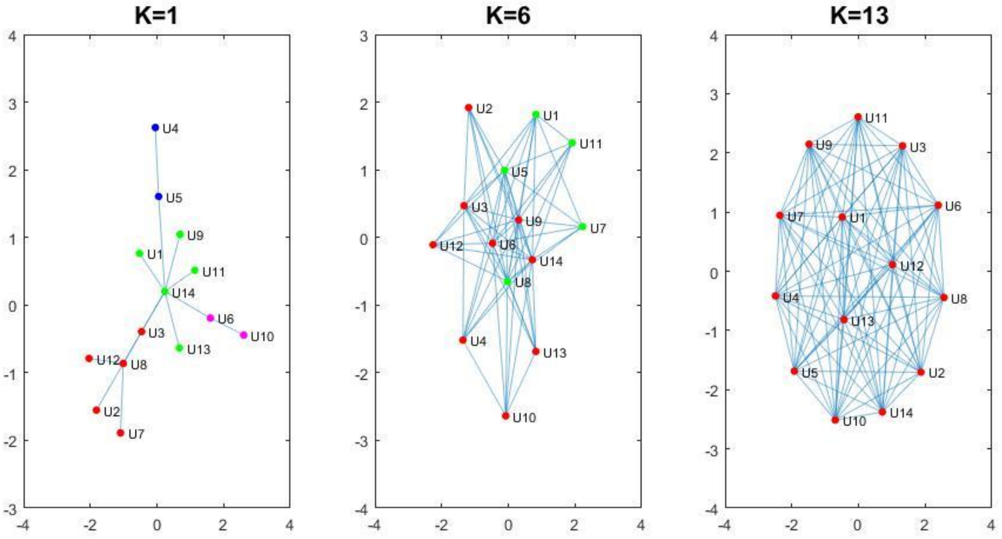


Figure 41. Communities detected when NGA algorithm is used. Node color represents the communities found in each graph and K represents average degree

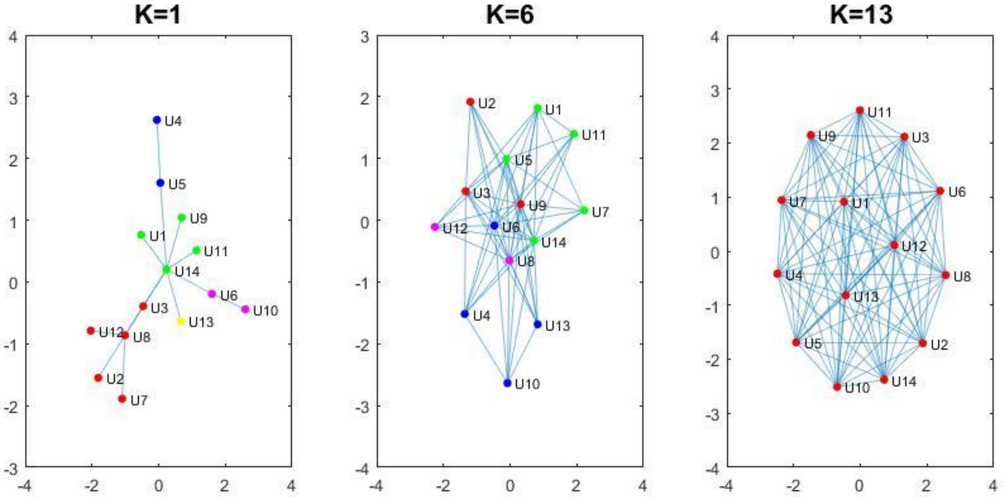


Figure 42. Communities detected when LeMartelot algorithm is used. Node color represents the communities found in each graph and K represents average degree

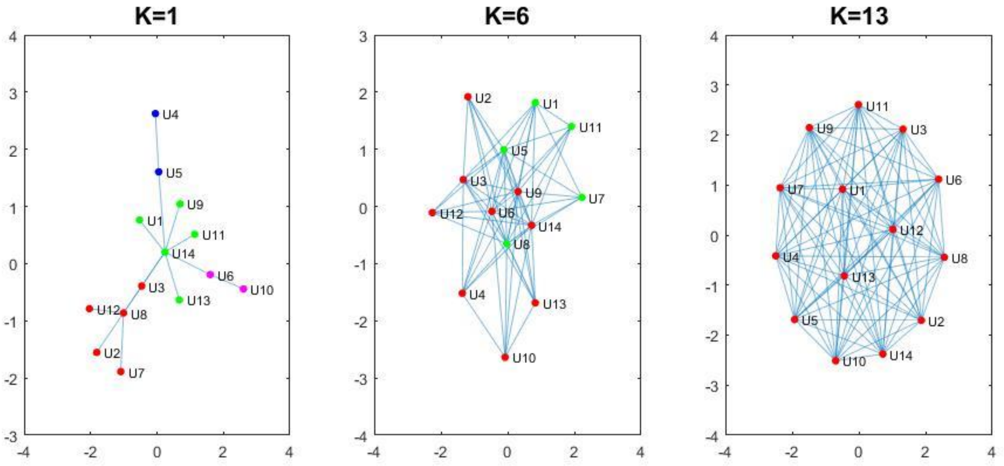


Figure 43. Communities detected when Danon algorithm is used. Node color represents the communities found in each graph and K represents average degree

Figure 40 to Figure 43 show an example of the influence of graph creation and community detection algorithms on the clustering result on the collected data. As can be observed in all cases, regardless of the community detection algorithm selected, for K equal to 13, all the users cluster in one community. When K is equal to 6, the communities found are different from one algorithm to another. Because of the low number of participants here, it is possible to compare the detected communities with actual communities. For this purpose, the actual communities of users are manually extracted, by looking to their movement patterns and frequently visited locations, and then compared to detected communities.

4.5.2.2 Effect of community detection algorithm on applied method

In the second scenario, in order to verify which algorithm will give the best result in terms of partition similarity, the effect of the community detection algorithms on the applied method is evaluated.

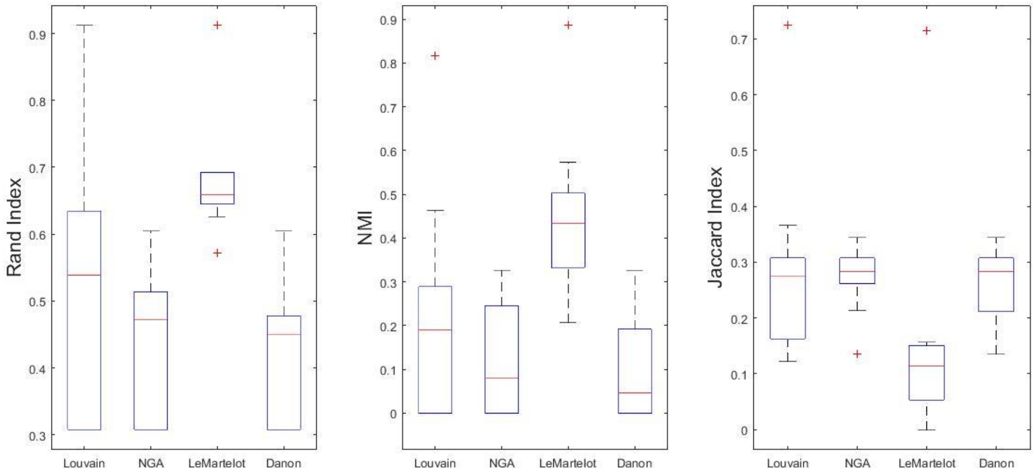


Figure 44. The effect of community detection algorithms on partition similarity (RI, NMI and JI)

As shown in Figure 44, the LeMartelot algorithm obtained the best partition similarity in both the RI and NMI, but not in the JI.

4.6 Conclusion and Discussion

In this chapter, a new method to detect communities of individuals was proposed and the advantages of applying community detection algorithm on time-series data is evaluated. Four different community detection algorithms were used to detect the community of individuals based on their collected sensor data. While only smart phone GPS sensor data was used in this analysis and could be seen as a limitation, the technique is easily applied within the context of an individual having more than one form of streaming sensor data. In order to evaluate the accuracy of the method, three different partition similarity indices were used to compare the detected communities with planted communities.

Chapter 5

5 Community Detection and Analysis Using Clustering Algorithms

This chapter demonstrates another method beyond graphs that can be used in the Community Detection component to detect communities. In addition, the case study used in this chapter provides a demonstration of an infrastructure based community as the sensor data comes from cars. Within this chapter, the community detection component of the proposed framework in Chapter 3 will be instantiated in a platform enabling community-based analytics. This platform finds vehicles with similar trajectories and groups them into communities. Vehicles which are grouped in the same communities may share rides with each other and decrease the total number of kilometers driven.

5.1 Introduction

As urban populations grow, cities need new strategies to maintain a good standard of living while enhancing services and infrastructure development [126]. A key area for improving city operations and spatial layout is the transportation of people and goods. While conventional transportation systems (i.e., fossil fuel-based) are struggling to serve the mobility needs of growing populations, they also present serious environmental threats. Alternative-fuel vehicles can reduce emissions that contribute to local air pollution and greenhouse gases as mobility needs grow. However, even if alternative-powered vehicles were widely employed, road congestion would still increase. This chapter investigates ridesharing as a mobility option to accommodate growing transportation needs and reduce overall congestion as well as the number of kilometers driven. The potential of ridesharing using personal vehicles in Changsha, China, is examined by reviewing mobility patterns of vehicles over two months. Big Data analytics identify ridesharing potential among these drivers by grouping vehicles according to their trajectory similarity in different communities. The approach includes five steps: data preprocessing, trip recognition, feature vector creation, similarity measurement and clustering. The potential reduction in kilometers driven through ridesharing among a specific group of drivers is calculated and discussed. Within the study area, ridesharing has the potential to reduce total kilometers driven by about 24%, assuming a maximum distance between trips of less than 10 kilometers, and a schedule time of less than 60 minutes. For a more conservative maximum trip distance of three kilometers and a passenger schedule time of less than 45 minutes, the reduction in traveled kilometers is more than 15% of total kilometers.

5.2 Factors Affecting Evaluation of Ridesharing Potential

In the analysis of ridesharing potential using driver mobility data, it is crucial to define what data is measured, how it is measured and how the data is analyzed. These parameters can greatly impact the findings and are often the reason for varied results in various studies. They are reviewed in the next two subsections.

5.2.1 Vehicle Trip Dataset

Large-scale data on vehicle mobility patterns in a city are needed in order to analyze ridesharing potential for reducing the overall demand for personal vehicles. This could include recorded location and time of day for all vehicles for a given time period (e.g., a day). The datasets used in ridesharing models vary depending on the following factors:

- Granularity of data (spatial and temporal): This often depends on the tools used to collect the data (e.g., cellphone, GPS systems, and social networking tools). In general, cellphone datasets, often in the form of Call Detail Records (CDRs), have less granular information in terms of user trajectories since they often record user information when users make calls or send text messages. For the purpose of Big Data collection in user mobility for ridesharing analysis, cellular data can be collected from network companies. Accuracy of such cellular data, often not specifically designed to indicate accurate location by using cellphone applications, is limited by the density of existing cellular towers in the area of user movements. In some cases, cellular telephone towers could cover a large area (up to several square kilometers) in rural areas, resulting in less accurate data. In contrast, GPS data rely on satellites and provide more accurate descriptions of user movements. The collection of cellular data from a larger number of users can provide data with acceptable accuracy but comparable to GPS-collected data. Data from online networks are also unable to reach high granularity, as they can only be collected when users post a geotagged message in a social network.

- Dataset size: This corresponds to the number of recorded trips over a period of time that affects the potential of ridesharing. In [127] the authors studied how the number of shareable trips in a given day varies as a function of the total number of recorded trips. In their study, in which the average number of daily-recorded trips in New York is approximately 400,000, the authors showed that, at approximately 100,000 trips, taxi ridesharing potential reaches its maximum theoretical value.

5.2.2 Data Analysis to Model Ridesharing

Once data on user mobility patterns are collected, extraction of suitable information and analysis to identify potential shared rides is a complex process consisting of several stages and dependent on several factors. Potential ridesharing opportunities are often presented as the fraction of individual trips that can be shared, sometimes referred to as shareability [127]. Many of the optimization challenges highlighted in [128] that arise when developing technology to support ridesharing and reviewed the relevant operations research models in this area.

5.2.2.1 Spatial and Temporal Constraints

The findings of user trip compatibility analyses are directly affected by the maximum allowed extra distance for each trip as a result of ridesharing as well as spatial (i.e., ride potential within a certain distance) and temporal (e.g., pick up and drop off within a time frame) constraints.

5.2.2.2 Number of Users Allowed to Share Rides

Some studies investigate the effect of the maximum number of rides to be shared in ridesharing potential. The authors in [129] found that as the limit on the number of shared rides increases, shareability potential also increases. It should be noted that an increase in the number of allowed shared rides is expected to increase extra travel distance and number of extra stops for each trip, two parameters that are often set to limited values in the models. Increasing the number of allowed shared rides would likely be ineffective in increasing shareability potential if these parameters are strictly kept at relatively low values. The authors in [129] found that, for three shared trips, the total saving in the total distance through ridesharing is 29% on average, with an average extra distance of 0.92 kilometers, while for two shared trips the saving is 18.2% with the average extra distance of 0.56 kilometers.

5.2.2.3 Trip Matching Algorithms: En-Route versus Origin-Destination Ridesharing

Another factor affecting the findings is the trip matching algorithms used in the analysis, and the ability of the model to capture en-route ridesharing (i.e., ride potential along trips).

5.2.2.4 Dynamic versus Static Ridesharing

In some models, it is assumed that trips are known in advance, which makes them suitable for carpooling applications but debatable for taxi ridesharing applications where opportunities are computed in real time. Taxi ridesharing requests arrive in real time and the algorithms used in evaluating such potential need to run large-scale studies that explore a wide range of scenarios through parameter sweeps. This often takes considerable computation time and, although many algorithms are capable of evaluating ridesharing potential among users, some are not able to evaluate such potential under the time constraints typically present in applications employed for connecting users. Thus, time constraints affect the calculated potential by the algorithms. In order to model the time-sensitivity of ridesharing potential, a time window is often used in the algorithms, outside of which ridesharing potential is not considered practical in real-time situations. Therefore, the potential for ridesharing is generally found to be lower in studies that account for this factor.

5.2.2.5 Factors Affecting Adoption of Ridesharing

In analyses of ridesharing potential, indirect factors that affect its adoption, such as passenger safety (i.e., riding with strangers) and privacy (i.e., disclosure of home and work addresses) are sometimes accounted for. Some studies focus on characterizing crowd mobility and activity patterns using information from social networks ([130-132]). The authors in [133] used online social network data to apply social constraints in their analysis of data for matching drivers (e.g., ridesharing among people who know each other). They found that, if users are willing to ride with friends of friends, the potential reduction is up to 31%, but if they are willing to ride only with people they know, the potential for ridesharing becomes negligible.

5.3 Objective: Estimation of Reductions in Driven Kilometers as a Result of Ridesharing

In the current study, the potential of ridesharing to reduce the number of kilometers driven is investigated. The trip GPS data of approximately 9,000 privately-owned vehicles in Changsha, China, are used. Ridesharing potential is identified according to trip origin and destination. The findings support the potential of ridesharing to improve congestion and local air quality.

5.4 Methods

This section introduces a proposed data-driven model that enables the analysis of historical location data to investigate the potential for ridesharing. There are several challenges related to this research, including: removal of outliers, noise and false data; investigation of the reliability of data; detection of misrepresented information in terms of location; feature selection; and clustering the data which significantly affects the findings. Figure 45 illustrates a data flow diagram for an analysis of ridesharing potential that consists of three steps of data processing: pre-processing, similarity detection and ridesharing recommendations.

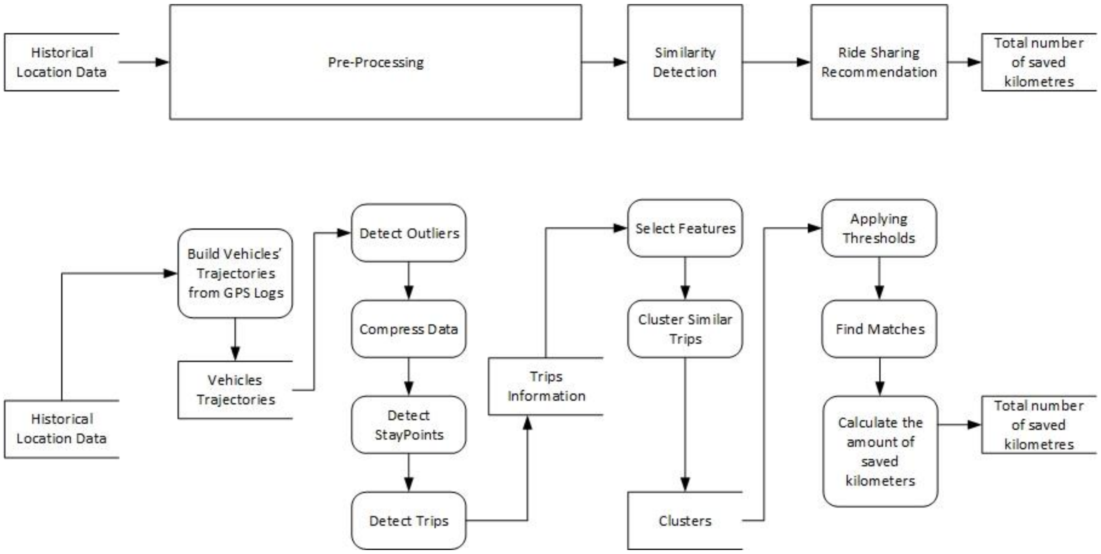


Figure 45. Data flow diagram for ridesharing

In this study, the vehicles' geographical locations (latitude and longitude) were collected using GPS monitoring systems installed in vehicles in Changsha, China (population 7 million). The historical data is processed to determine possible similar rides that could be shared. The potential number of kilometers saved by adopting ridesharing is calculated.

It should be noted that ridesharing in the current analysis is short-distance, static and on a daily basis. It is also assumed that wherever matching trips exist, the car that corresponds to the longest trip is selected as the one that provides the ride to others, and is the one setting the origin and destination of the shared trip. Passengers of the cars corresponding to the other trips (i.e., riders) are expected to walk the last part of their trip (also called the last mile) from the driver's destination to theirs.

5.4.1 Pre-Processing

5.4.1.1 Trajectory Representation and Location History Modeling

As depicted in Figure 45, spatial-temporal trajectories are first built from GPS logs. The data is retrieved from the database for each vehicle and transformed into a series of chronologically ordered points: for example, P1→P2→P3→…→Pn. Each trajectory point consists of a timestamp, geospatial coordinates (latitude, longitude) and the speed of the vehicle.

Data pre-processing is a crucial step as data collection is often loosely controlled, resulting in outliers, noise, and missing information. Thus, to reduce the complexity of data analysis and program execution time, the following data pre-processing and representation steps were applied.

5.4.1.2 Noise Filtering and Outlier Detection

The first step in data pre-processing is noise filtering and outlier detection, which searches for abnormalities in trajectories. Outliers in trajectories can be a point or series of points that are significantly different from other points. For instance, an outlier can be a point that is far from other points and out of possible vehicle reach within the regulated speed and time. An outlier can also be a point of observation that does not conform to the expected pattern. In this study, a mean filter [134] was used to detect the noise and outlier. For point Pz in a vehicle's trajectories, a true value is the mean of the position of Pz and the n-1 predecessor, thus the mean filter can be a sliding window covering the n adjacent values of Pz:

$$\sum_{i=(z-n+1)}^{z} P_i \tag{16}$$

where n is the size of the sliding window for the mean filter.

5.4.1.3 Compression

While vehicle locations can be constantly sampled and communicated, a high rate of sampling can result in excessive communication overhead, computing and data storage. It is also important to consider that when a vehicle is waiting at a traffic light, or delayed in congestion, its location does not change for a while, but it is still continuously sampled. To decrease the volume of data and improve the performance of data processing, the points from trajectories for which there is no updated information are removed.

5.4.1.4 Stay Point Detection

An important part of the analysis is to detect stay points because they can be used in trajectory segmentation and trip detection. Stay points denote locations, such as parking lots, where vehicles stay for more than five minutes. There are two different types of stay point: a single point location where a vehicle remains stationary; and when a vehicle location is updated but there is no notable change in the location. In this study, both types of stay point are detected.

5.4.1.5 Trip Detection

To group similar trips, a trajectory first needs to be divided into different trips. Segmenting trajectories into trips helps to reduce computation cost and to facilitate deeper study into vehicle trajectories as well as to identify more potential ridesharing options. In this study, trips are detected according to time interval and stay points. For example, if the time interval between two consecutive points in a vehicle trajectory is larger than a defined threshold, the vehicle trajectory can be divided into two trips. Furthermore, stay points can divide a trajectory into two different segments or trips.

5.4.2 Similarity Detection

The main purpose of the present analysis is to detect similar rides and mark them for potential ridesharing. In this step, clustering detects similar trips and groups them together.

5.4.2.1 Feature Selection

As different trips contain different properties such as length, number of points, and sampling rate, it is difficult to use trip properties for clustering. To solve this issue, useful features can be selected from each trip and uniformly presented. For the purposes of this study, the start time, end time, origin, destination and length of each trip are used to describe the trip features and are represented as a vector.

5.4.2.2 Clustering

Clustering in this analysis is the process of grouping similar trips. The trips inside a group are share greater similarity than other trips that are placed in other groups or clusters. The distance between the trips is measured by the distance between vectors. Clustering attempts to minimize the distance between the trips inside of each cluster and to maximize the distance between trips outside of each cluster.

One of the most commonly used algorithms for clustering is the k-means [135], which is an iterative clustering algorithm that partitions n observations into a number of clusters (k) that are selected before the algorithm starts. In this study, k-means is used for grouping similar trips. It randomly chooses k initial cluster centers and calculates the distance of the centroid in each cluster to all the trips, then assigns each trip to the group with the closest centroid. Subsequently, in order to find the new centroid, k-means calculates the average distance between trips inside of each cluster and the cluster centroid. It repeats these steps until the cluster members do not change.

For the purpose of measuring the similarity between trips and their centroids in this study, multiple similarity functions, such as Euclidean, Cosine, City block and Correlation [136], were used. For each of these functions, the distance was calculated based on the following equations:

$$\text{Euclidean: } d(x,c) = \sqrt{\sum_{i=1}^{p}(x_i - c_i)^2} \tag{17}$$

$$\text{City block: } d(x,c) = \sum_{i=1}^{p}|x_i - c_i| \tag{18}$$

$$\text{Cosine:} \qquad d(x,c) = 1 - \frac{xc'}{\sqrt{(xx')(cc')}} \tag{19}$$

$$\text{Correlation: } d(x,c) = 1 - \frac{(x-\bar{x})(c-\bar{c})'}{\sqrt{(x-\bar{x})((x-\bar{x})'}\sqrt{(c-\bar{c})(c-\bar{c})'}} \tag{20}$$

where $\bar{x} = \frac{1}{p}\left(\sum_{j=1}^{p} x_j\right)\overline{1_p}$ and $\bar{c} = \frac{1}{p}\left(\sum_{j=1}^{p} c_j\right)\overline{1_p}$

where p is the dimension, x is an observation or feature vector for a trip, c is a centroid and ($\overline{1_p}$) is a row vector of p ones.

5.4.3 Ridesharing Recommendations

While clustering partitions similar trips into groups, it does not guarantee that all the trips inside each group have the potential for ridesharing. There still remain limitations for ridesharing, such as: the maximum distance between the trip start and end points; the maximum user schedule time; the maximum number of passengers who can share the ride; and the minimum length for which two users opt to travel together. In this step, such thresholds are considered for each cluster. The potential trips that could be shared are estimated.

5.5 Experimental Analysis

In this section, the performance of the approach used in this study is demonstrated using the GPS location records of 8,900 privately-owned vehicles in Changsha, China. In the experiments, the effect of different similarity functions, along with a different number of clusters on the clustering algorithm, are examined in order to determine the best option for ridesharing. The effect of maximum schedule time and maximum distance between the trip start and end points is also examined. The results show that a Euclidian similarity function with 11,000 clusters achieves the best performance and that there is no notable change on the total number of saved kilometers if the maximum schedule time is increased to more than one hour and the maximum distance between the trip start points and endpoints is increased to more than six kilometers.

5.5.1 Experimental Setup

The historical data of every vehicle was sampled every 10 minutes and stored in a database. Thus, the historical dataset that was included in this study was also sampled every 10 minutes, totaling 65,940,000 records spanning 89 days from February to April 2013. In an ideal situation, each vehicle would create 144 records per day, resulting in 114,062,400 for 8,900 vehicles for 89 days. However, the applied monitoring system did not collect data from vehicles that remained stationary for more than 12 hours. Moreover, there is typical data loss which can be attributed to a variety of reasons. For example, monitoring data was wirelessly communicated to the monitoring platform using cellular GPRS networks, which is error-prone due to the nature of the wireless channel that introduces data loss, delay, and retransmissions.

The experiments ran on a server with Intel 6 cores Xeon E5649 2.53GHz processor, 32 GB RAM and a Windows server 2016 operating system running MATLAB R2016b. MATLAB was used as the programming environment for the experiments. A MATLAB parallel computing toolbox was also used to gain maximum benefit from the multiple cores inside the server processor. The toolbox enabled the use of the full power of the multicores by executing the program on multiple threads.

5.5.2 Ridesharing in 24 Hours

To demonstrate the performance of the approach, the first day (24 hours) of the dataset, which contains 1,080,224 records, was selected. This included travel typical of a weekday. The total traveled distance on this day was 201,890 kilometers while the total number of detected trips was 20,018, resulting in an average trip length of 10.53 kilometers. Figure 46 shows the total hourly travel distance driven by the vehicles for 24 hours on the first day of the dataset. Figure 47 indicates the trip start points for 24 hours on an actual map. When rides are shared, it is assumed that the maximum capacity of each vehicle, including the driver, is four passengers. In addition, it is assumed that sharing rides that are shorter than two kilometers results in excessive detouring and provides negligible benefits in terms of reduction in overall trip kilometers. As a result, trip data corresponding to such trips were excluded from the experiment.

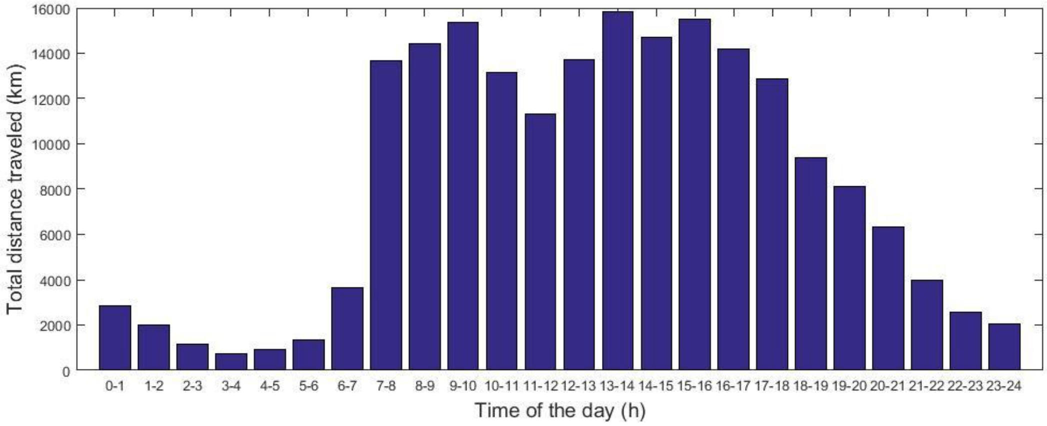


Figure 46. Total hourly distance driven by vehicles over a 24 hours period

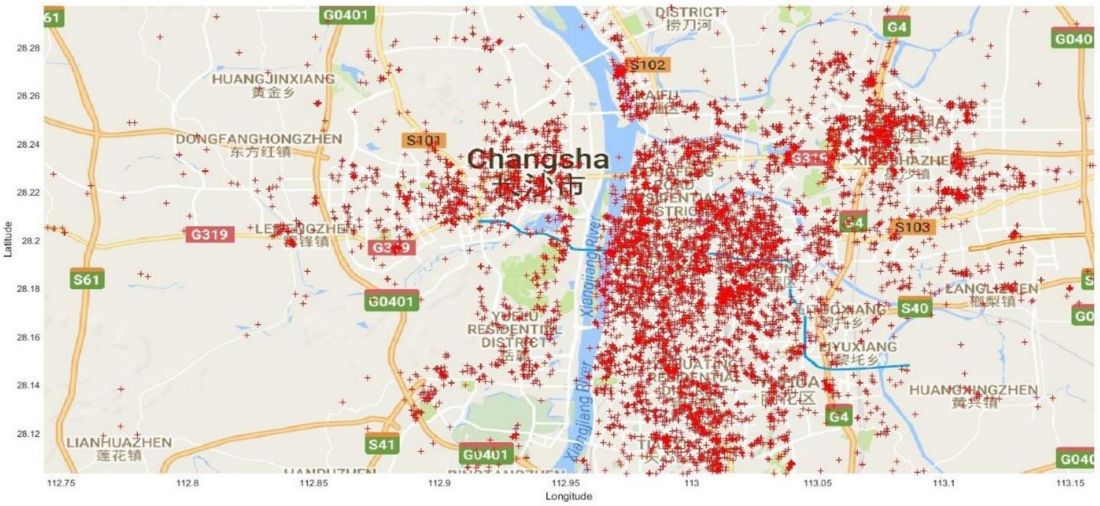


Figure 47. Trip start points (red dots) over a 24 hours period

5.5.2.1 Effect of the Similarity Function on Ridesharing

5.5.2.1.1 Scenario 1

In the first scenario, the effect of different similarity functions and maximum schedule time on ridesharing potential was evaluated. Table 7 shows the values assigned for the simulation set-up parameters for this first scenario. It was assumed that the number of clusters is constant and equal to 8,000 clusters. To match trips with ridesharing potential, the maximum time that passengers can wait for a ride (here referred to as schedule time) and the maximum allowable distance between trip origins and destinations (also here referred to as trip distance) are set. In this scenario, the maximum schedule time varies between 5 and 180 minutes, and the maximum distance between trips is set to two kilometers. It is established that the Euclidean and City block similarity functions result in the highest values of total saved kilometers (Figure 48 (a)) and the total number of saved trips (Figure 48 (b)) if the maximum schedule time is less than an hour. If the maximum schedule time is more than 60 minutes, the City block similarity function indicates higher values in total saved kilometers and the saved number of trips compared to other functions. Both the Euclidean and City block similarity functions have better results in terms of saved kilometers because they act better on the data that can be represented as points in a Euclidean space. The cosine similarity measures the angle between two vectors. While it is a suitable candidate for multi-feature vectors, it did not perform well for the small number of features' vectors. The correlation similarity function is also only suitable for high-dimensional data which is not the case in this study.

Table 7. Simulation set-up parameters (scenario 1)

Description	Value
Similarity Function	Variable
Maximum distance between trips (Kilometers)	2
Number of clusters	8000
Maximum schedule time (Minutes)	5–180
Total trip length (Kilometer)	210,890
Total number of trips	20,018

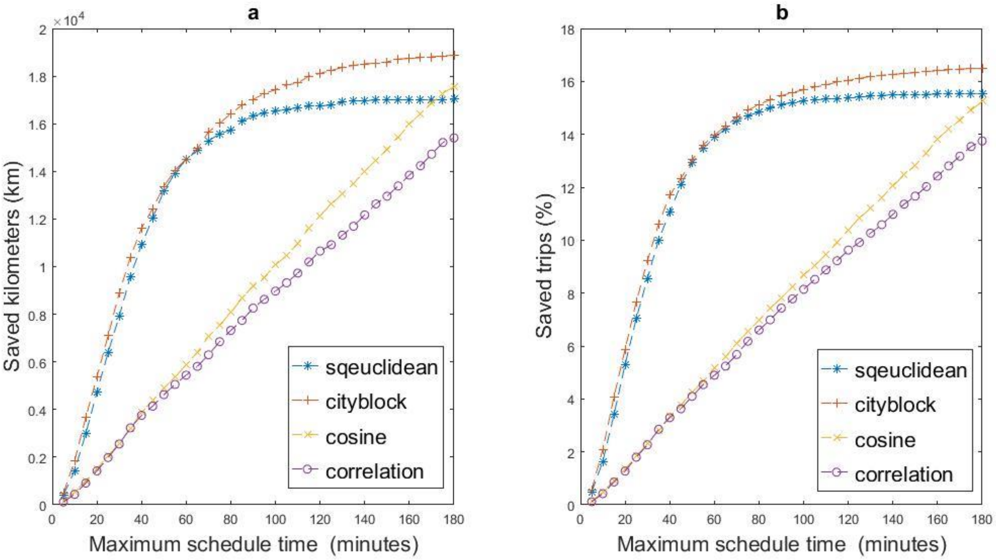


Figure 48. Effect of similarity function on (a) Saved kilometers (b) Number of saved trips

5.5.2.1.2 Scenario 2

In this scenario, the effect of the similarity function and the maximum distance between trips on ridesharing potential is evaluated. Table 8 shows the values assigned for the set-up parameters for the second scenario. The number of clusters is assumed to be a constant and equal to 8,000 clusters. The maximum schedule time is set to 40 minutes, and the maximum distance between trips is a variable between 1 and 20 kilometers. It is established that the Euclidean and city block similarity functions result in higher values of total saved kilometers (Figure 49 (a)) and the number of saved trips (Figure 49 (b)) compared to the cosine and correlation functions. As can be observed in Figure 36, there is no improvement in ridesharing potential if the distance between the trips is more than six kilometers due to the decrease in similarity among trips when the distance between them is increased. Ultimately, when the distance is more than six kilometers, there is no similar trip available for matching inside each cluster.

Table 8. Simulation set-up parameters (scenario 2)

Description	Value
Similarity Function	Variable
Maximum distance between trips (Kilometers)	1–20
Number of clusters	8,000
Maximum schedule time (Minutes)	40
Total trip length (Kilometer)	210,890
Total number of trips	20,018

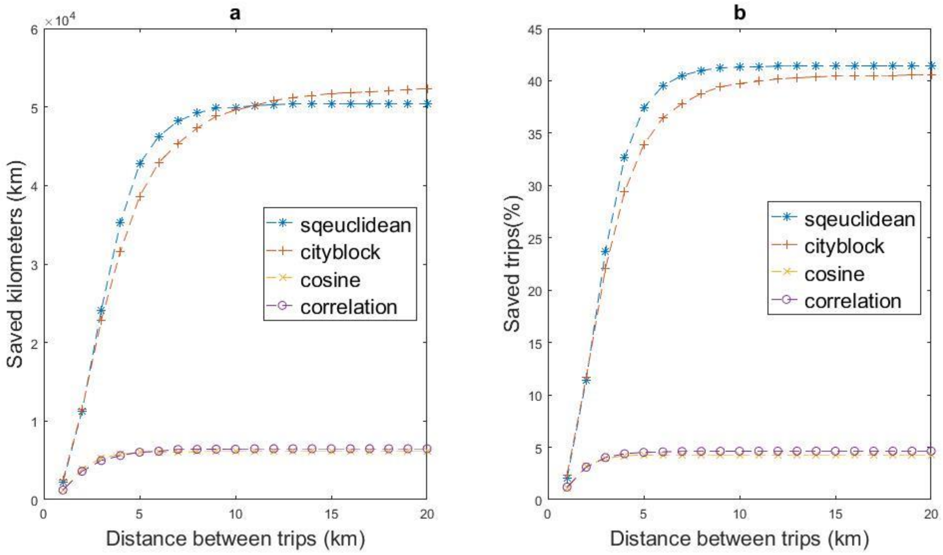


Figure 49. Effect of similarity function on (a) saved kilometers (b) number of saved trips

5.5.2.1.3 Scenario 3

In the third scenario, the effect on ridesharing potential of the similarity function and the number of clusters was investigated. Table 3 shows the values assigned for the set-up parameters for this scenario for which it is assumed that the number of clusters is a variable between 1,000 and 15,000. The maximum schedule time is kept to 40 minutes, and the maximum distance between trips is kept to three kilometers. The highest values of saved kilometers and the total number of saved trips are achieved with the Euclidean similarity function when the number of clusters is approximately 11,000 (Figure 50). As Figure 50 depicts, increasing the number of clusters to more than 11,000 does not increase the total number of saved kilometers. This can be explained by the decrease in the number of similar trips inside of each cluster as the number of clusters is increased.

Table 9. Simulation set-up parameters (scenario 3)

Description	**Value**
Similarity Function	Variable
Maximum distance between trips (Kilometers)	3
Number of clusters	1000–15,000
Maximum schedule time (Minutes)	40
Total trip length (Kilometer)	210,890
Total number of trips	20,018

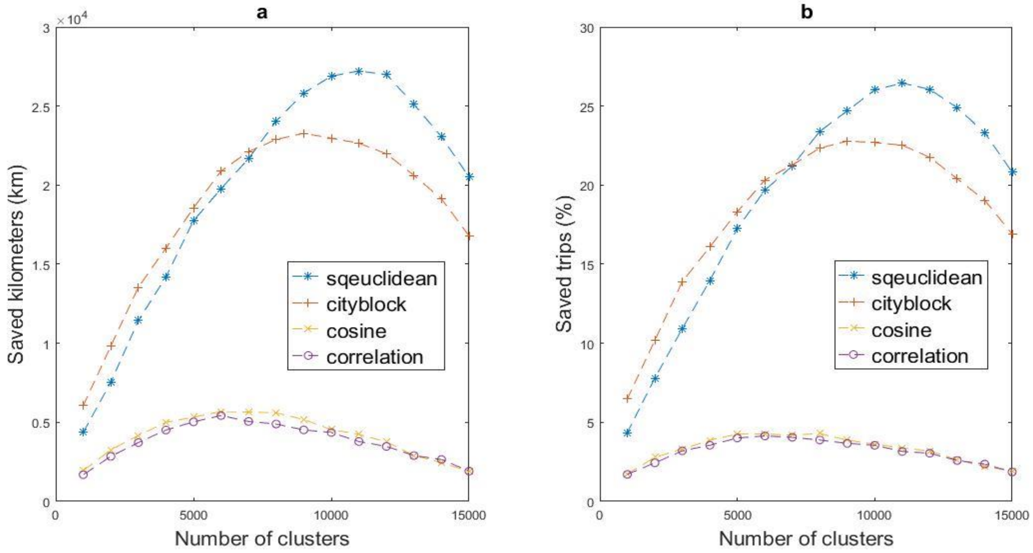


Figure 50. Effect of similarity function on (a) saved kilometers (b) number of saved trips

5.5.2.2 Scenario 4: Effect of the Number of Clusters and Schedule Time on Ridesharing

This scenario explores the effect on ridesharing potential of changing the number of clusters and schedule time. Table 10 shows the values assigned for the set-up parameters for this scenario, for which it is assumed that the number of clusters is a variable between 1,000 and 15,000, that the maximum schedule time is a variable between 5 and 180 minutes, and that the maximum distance between trips is a constant, equal to three kilometers. The largest reduction in traveled kilometers is achieved with 11,000 clusters if the maximum schedule time is less than an hour (Figure 51).

Table 10. Simulation set-up parameters (scenario 4)

Description	Value
Similarity Function	Euclidean
Maximum distance between trips (Kilometers)	3
Number of clusters	1,000–15,000
Maximum schedule time (Minutes)	5–180

Total trip length (Kilometer)	210,890
Total number of trips	20,018

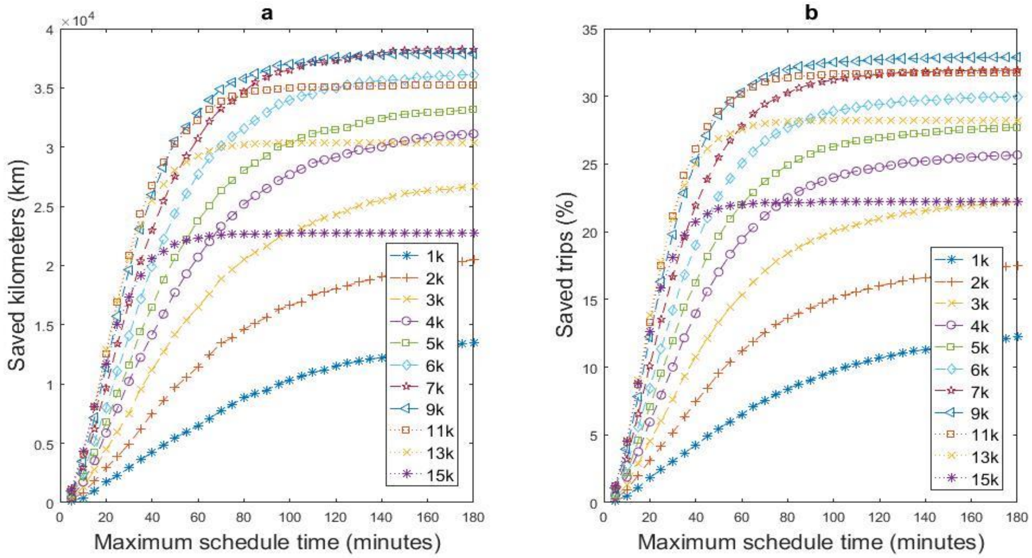


Figure 51. Effect of number of clusters on (a) saved kilometers (b) number of saved trips

5.5.2.3 Scenario 5: Effect of Maximum Trip Distance and Schedule Time on Ridesharing

In scenario 5, the effect of trip distance and schedule time on ridesharing potential is investigated. Table 11 shows the set-up parameters for this case. As determined in the previous scenarios, the highest values of saved kilometers are achieved using the Euclidean similarity function with 11,000 clusters. In this scenario, the number of clusters is maintained at 11,000 and Euclidean distance is used for the similarity function. The results show a saving of more than 15% on total travel distance (Figure 52 (a)) and more than 30% on the number of trips (Figure 52 (b)) if the maximum distance between trips is three kilometers and the maximum schedule time is 45 minutes. It is observed that if the maximum schedule time is increased to more than 60 minutes, there is no significant change in the number of saved kilometers. As a consequence, the maximum time lag between the trips inside any cluster is 60 minutes. Furthermore, by increasing the maximum distance between the trips to more than six kilometers, there is no change in the total number of saved kilometers.

Table 11. Simulation set-up parameters (scenario 5)

Description	Value
Similarity Function	Euclidean
Maximum distance between trips (Kilometers)	1–10
Number of clusters	11,000
Maximum schedule time (Minutes)	5–180
Total trip length (Kilometers)	210,890
Total number of trips	20,018

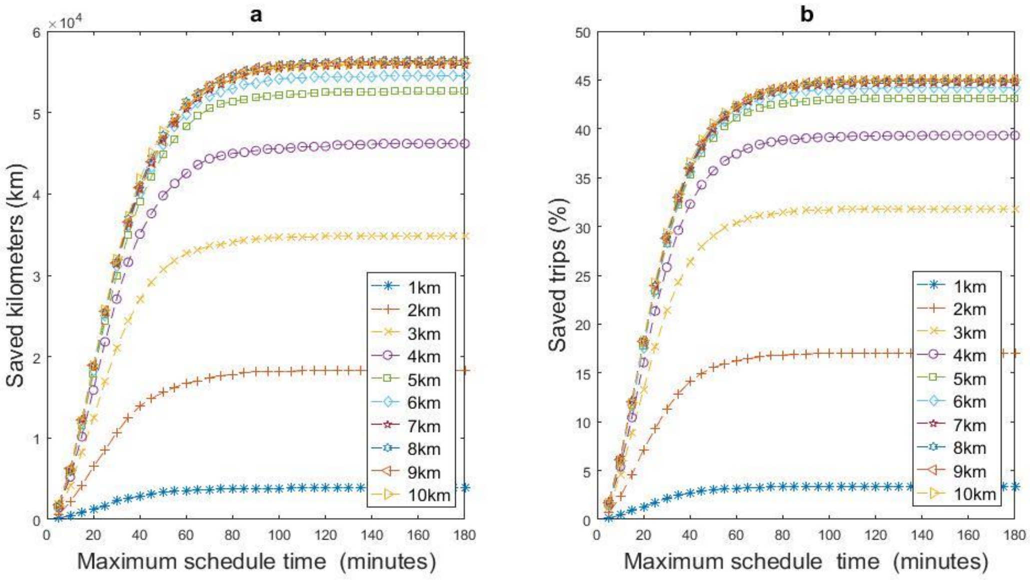


Figure 52. Effect of trip distance on (a) saved kilometers (b) number of saved trips

5.6 Conclusion and Discussion

Adoption of ridesharing among passenger vehicles in Changsha, China, as a potential strategy to reduce kilometers driven is investigated. Historical GPS data of privately-owned vehicles in Changsha, China, are collected and used in an algorithm that is developed to match riders with close temporal and spatial origin and destinations. The developed algorithm is capable of estimating the number of kilometers that are reduced among users if ridesharing is adopted.

The results show the potential of ridesharing to reduce total traveled distance depends on the users' tolerance towards changes to their original trip route and departure time.

As shown in previous studies, the size of the dataset can affect the potential for ridesharing among users. Therefore, the results of the current study are dependent on the size of the dataset used to identify potential ridesharing opportunities among users. A larger dataset (i.e., more participants) would match more riders with ridesharing. As a result, the estimated traveled distance reduction from ridesharing adoption in Changsha, China, are expected to be higher with a larger pool of participants.

While the quantitative results of this analysis are specific to the population under study, they provide useful insights into the potential of ridesharing for improving air quality and reducing emissions associated with climate change. Changsha, China, is one of several cities around the world that use personal vehicles as a reliable mode of transportation. The methods used in this study to evaluate ridesharing potential for reducing traveled kilometers in Changsha can be used in future similar studies on other cities that partially or fully rely on personal vehicle transportation. Analysis of the current demand for transportation and projection of future trends is a key task in planning for sustainable transportation modes, such as ridesharing, that are potentially able to meet future demand.

Within the study area, ridesharing has the potential to reduce total kilometers driven (210,890 kilometers) by about 24% (51,087 kilometers) and vehicle trips (20,018 trips) by approximately 40% (8480). This maximum potential assumes a maximum distance between trips of less than 10 kilometers, and a schedule time of less than 60 minutes (Figure 52). If a more conservative maximum distance of two kilometers between trips and a schedule time of less than 40 minutes is selected, the total distance traveled reduces by 7% and the total number of trips decreases by 14%.

While, in this study, only one day was selected for data analysis and for investigating the potential of ridesharing, a future study could analyze the stability of data over a longer timeframe in order to determine whether the communities detected for a specific day are stable during a longer period. The new study could analyze the data for multiple weeks in order to ascertain the presence of a stable pattern in the detected communities. Such a study has the potential to prove that ridesharing has benefits not only for a specific day but over a long period.

It must be noted that, although the findings of this study illustrate the potential of ridesharing for reducing driven kilometers, its adoption by users still faces challenges such as passenger safety, privacy and liability. Furthermore, the success of web-based applications in connecting potential shared rides is dependent on the number of users. In terms of regulations, they compete with existing regulated taxi companies. Such limitations need to be further analyzed in order to determine solutions to overcome these challenges.

Chapter 6

6 Conclusion and Future Work

Cities have changed considerably over the last century. As populations have increased, cities are faced with a scarcity of resources that requires a re-evaluation of approaches to management. Within the next 35 years, the world's cities will double in size, therefore smart cities are needed in order to manage this sharp increase. This study has proposed a new architecture and framework for smart cities which would enable community data collection and analysis. Furthermore, this work has suggested a new method which takes real time data from individual smartphone sensors and groups them according to their similarities and common interests. Identifying communities of individuals and grouping people allows for the establishment of functionality and interactions between the network members in order to predict their relations and to infer missing attributes and features. Furthermore, detecting a community of common interests allows to provide better services for those who are interested in similar things. Finally, when their communities and common interests are known, many applications in different domains can be developed to provide users with better services.

This final chapter first concludes the research contribution of this study to community-oriented analysis. Potential issues for future work which have not been included in this study are then identified and discussed. Finally, concluding remarks are presented.

6.1 Summary and contribution

Chapter 2 presents an extensive study of the IoT, the smart city and community sensing. IoT vision, IoT enabling technologies and smart city applications are reviewed in Sections 2.1.1 to 2.1.3. Architecture and middleware for the smart city and the IoT are compared in Section 2.14. More than 23 different middleware are reviewed and their functionalities compared in Tables 2 and 3. Cloud computing, device capability abstraction and context-aware computing for the IoT are reviewed and discussed in Sections 2.1.5 to 2.1.7. Additionally, Section 2.2 focuses on community sensing and mining in order to show the benefits of community analysis in smart cities.

A community-oriented architecture for the smart city is proposed in Chapter 3 together with the design for a framework that enables community-analysis. A cloud-based general architecture for smart cities, which allows community service providers, city management and citizens to access real time data that has been gathered from the city through IoT in order to ensure the provision of essential services and improved quality of life for city residents, is suggested in Section 3.1. The community-oriented framework, as presented in Chapter 3.2, suggests that community analysis for the smart city can solve issues such as heterogeneity in data collection, data quality and Big Data management. A community detection component detects the communities of people and groups them according to their similarities and common interests. This framework can benefit from various models in order to build smart city applications for urban planning, sustainable communities, transportation, public health, public security, and commerce.

Chapter 4 provides a definition of community and discusses the benefits of detecting communities. A method to detect and analyze communities of individuals using graphs, based on the sensor data collected from individual smartphones, is proposed in Section 4.4. This method can be used as an implementation of community detection and the data aggregation component of the proposed framework in Chapter 3. The proposed method acquires time series data from multiple sensors and converts it to correlation graphs by applying various similarity functions, then integrates the sensor correlation graphs and creates a user correlation network. In the final stage, the method can identify the communities of the corresponding correlation network by applying different community detection algorithms. The performance of the proposed method has been evaluated in Section 4.5 by running a case study that collected GPS data from multiple users and determined the communities of those users according to their movement patterns.

A method for community detection by using a clustering algorithm is proposed in Chapter 5. Although Chapter 4 used graphs to detect communities, this chapter used feature clustering to identify the communities of common interest. The proposed method for community detection was evaluated by running a case study that identified the potential for ridesharing from personal vehicles in Changsha, China, by reviewing mobility patterns of approximately 8,900 privately-owned vehicles over a two-month period. Big Data analytics identified ridesharing potential among these drivers by grouping vehicles into different communities according to their trajectory similarity. The results in Section 5.5 show that the potential of ridesharing to reduce total traveled distance varies significantly by the users' tolerance towards changes to their original trip route and departure time. Within the study area, ridesharing has the potential to reduce total kilometers driven by about 24% and vehicle trips by approximately 40%.

6.2 Future work

Several issues require further investigation, the most important of which include the following:

- Community analysis can be considered as a service to the application layer. Any application can use this service in order to detect communities of common interest.
- Different components of the proposed architecture, such as privacy management, data quality and task assigning components, can be implemented.
- Classification methods and algorithms can be combined to the architecture and will be used with clustering algorithms to analyze the communities of individuals.
- Overlapping communities can be detected and added to the proposed community detection method.
- Different similarity functions and community detection can be studied and added to the proposed method.
- Distinctive features of the communities can be investigated. The community detection method can derive benefits from them.

Appendices

Appendix 1: REB Application Approval

Date:	**September 19, 2016**
To:	**Khalil El-Khatib, Roozbeh Jalali**
From:	**Shirley Van Nuland, REB Chair**
Title:	**Community-Oriented Architecture for Smart Cities**
Decision:	**APPROVED**
Current Expiry:	**September 01, 2017**
REB File#:	**14059**

The University of Ontario, Institute of Technology Research Ethics Board (REB) has reviewed and approved the research proposal cited above. This application has been reviewed to ensure compliance with the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS2 (2014)) and the UOIT Research Ethics Policy and Procedures. You are required to adhere to the protocol as last reviewed and approved by the REB.

Continuing Review Requirements (forms can be found on the UOIT website):

- **Renewal Request Form**: All approved projects are subject to an annual renewal process. Projects must be renewed or closed by the expiry date indicated above ("Current Expiry"). Projects not renewed within 30 days of the expiry date will be automatically suspended by the REB; projects not renewed within 60 days of the expiry date will be automatically closed by the REB. Once your file has been formally closed, a new submission will be required to open a new file.
- **Change Request Form**: Any changes or modifications (e.g. adding a Co-PI or a change in methodology) must be approved by the REB through the completion of a change request form before implemented.
- **Adverse or Unexpected Events Form**: Events must be reported to the REB within 72 hours after the event occurred with an indication of how these events affect (in the view of the Principal Investigator) the safety of the participants and the continuation of the protocol (i.e. un-anticipated or un-mitigated physical, social or psychological harm to a participant).
- **Research Project Completion Form**: This form must be completed when the research study is concluded.

Always quote your REB file number (/**14059**) on future correspondence. We wish you success with your study.

REB Chair
Dr. Shirley Van Nuland
shirley.vannuland@uoit.ca

Ethics and Compliance Officer
researchethics@uoit.ca

Appendix 2: Experiment's Letter of Invitation

Hello

My name is Roozbeh Jalali and I am a PhD student working under the supervision of Dr. Khalil El-Khatib and Dr. Carolyn McGregor in the Faculty of Business and Information Technology at UOIT. The reason that I am contacting you is that we are conducting a study that automatically determines communities of individuals based on their physiological information and trajectory. We are currently seeking volunteers from UOIT and outside of the campus as participants in this study.

Participation in this study involves coming into the laboratory where you will be asked to download and install a smart-phone application and be provided with a headband. We will collect your location data from your smartphone sensor and physiological data from the brain sensing headband for a month and you will be asked to return the headband after the data collection period.

Participation in this study will take approximately one month.

As a token of appreciation for your time commitment, at the end of study your name will be entered in a draw for a Fitbit. I would like to assure you that this study was approved by the UOIT Research Ethics Board [REB # 14059] on [insert date]. However, the final decision about participation is yours.

If you are interested in participating, please contact me at Roozbeh.jalali@uoit.ca and I will then send a confirmation email indicating that you have been signed up, along with further information concerning the time and location of the study.

Sincerely

Roozbeh Jalali

References

[1] G. K. Heilig, "World urbanization prospects: the 2011 revision," *United Nations, Department of Economic and Social Affairs (DESA), Population Division, Population Estimates and Projections Section, New York*, 2012.

[2] R. Jalali, K. El-khatib, and C. McGregor, "Smart city architecture for community level services through the internet of things," in *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, 2015, pp. 108-113.

[3] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, 2011, pp. 1028-1031.

[4] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter cities and their innovation challenges," *Computer*, vol. 44, pp. 32-39, 2011.

[5] (2016). *Population, urban percent*. Available: http://pardee.du.edu/

[6] K. Ashton, "That 'internet of things' thing," *RFiD Journal*, vol. 22, pp. 97-114, 2009.

[7] D. L. Brock, "The electronic product code (epc)," *Auto-ID Center White Paper MIT-AUTOID-WH-002*, 2001.

[8] I. Strategy and P. Unit, "ITU Internet Reports 2005: The internet of things," *Geneva: International Telecommunication Union (ITU)*, 2005.

[9] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, pp. 2787-2805, 2010.

[10] G. s. Gartner, "Hype Cycle for Emerging Technologies Maps the Journey to Digital Business (2014)," ed, 2015.

[11] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, pp. 171-209, 2014.

[12] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50-58, 2010.

[13] (2015). *BUTLER*. Available: http://www.iot-butler.eu/

[14] (2015). *FIND*. Available: http://www.nets-find.net/

[15] V.-M. Scuturici, S. Surdu, Y. Gripay, and J.-M. Petit, "UbiWare: Web-based dynamic data & service management platform for AmI," in *Proceedings of the Posters and Demo Track*, 2012, p. 11.

[16] R. Rosso, G. Munaro, O. Salvetti, S. Colantonio, and F. Ciancitto, "CHRONIOUS: an open, ubiquitous and adaptive chronic disease management platform for chronic obstructive pulmonary disease (COPD), chronic kidney disease (CKD) and renal insufficiency," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010, pp. 6850-6853.

[17] J. Barbarán, C. Bonilla, J. Á. Dianes, M. Díaz, and A. Reyna, "Simulating SMEPP middleware," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008, p. 23.

[18] C. Sarkar, A. Uttama Nambi SN, R. Prasad, A. Rahim, R. Neisse, and G. Baldini, "DIAT: A Scalable Distributed Architecture for IoT," 2012.

[19] (2015). *COMPOSE*. Available: http://www.compose-project.eu/

[20] P. Newman, "The environmental impact of cities," *Environment and Urbanization,* vol. 18, pp. 275-295, 2006.

[21] P. C. Annez and R. M. Buckley, "Urbanization and growth: setting the context," *Urbanization and growth,* p. 1, 2009.

[22] J. Conway-Beaulieu, A. Athaide, R. Jalali, and K. El-Khatib, "Smartphone-based Architecture for Smart Cities," in *Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, 2015, pp. 79-83.

[23] R. Jalali, A. Dauda, K. El-Khatib, C. McGregor, and C. Surti, "An architecture for health data collection using off-the-shelf health sensors," in *Medical Measurements and Applications (MeMeA), 2016 IEEE International Symposium on*, 2016, pp. 1-6.

[24] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, 2014, pp. 593-598.

[25] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? A measurement study on GreenOrbs," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 24, pp. 1983-1993, 2013.

[26] H. Yue, L. Guo, R. Li, H. Asaeda, and Y. Fang, "DataClouds: Enabling Community-based Data-Centric Services over Internet of Things," 2014.

[27] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz*, et al.*, "Smart cities of the future," *European Physical Journal-Special Topics*, vol. 214, p. 481, 2012.

[28] S. C. Mukhopadhyay and N. Suryadevara, *Internet of Things: Challenges and Opportunities*: Springer, 2014.

[29] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems,* vol. 29, pp. 1645-1660, 2013.

[30] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi*, et al.*, "Internet of things strategic research roadmap," *O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, et al., Internet of Things: Global Technological and Societal Trends,* vol. 1, pp. 9-52, 2011.

[31] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks,* vol. 52, pp. 2292-2330, 2008.

[32] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, *et al.*, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Computing Surveys (CSUR)*, vol. 48, p. 7, 2015.

[33] A. Zanella, N. Bui, A. P. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal,* 2014.

[34] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, "Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype," in *Proceedings of the 4th workshop on Embedded networked sensors*, 2007, pp. 13-17.

[35] X. Wang, M. S. Gerber, and D. E. Brown, "Automatic crime prediction using events extracted from twitter posts," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, ed: Springer, 2012, pp. 231-238.

[36] C. Hargood, D. Michaelides, M. Weal, V. Pejovic, M. Musolesi, L. Morrison, *et al.*, "The UBhave Framework: Developing Dynamic Mobile Applications for Digital Behavioural Interventions."

[37] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated GPS traces," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 2, 2011.

[38] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, *et al.*, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, 2009, pp. 349-350.

[39] E. L. Glaeser and B. Sacerdote, "Why is there more crime in cities?," National Bureau of Economic Research1996.

[40] T. Bennett, K. Holloway, and D. P. Farrington, "Does neighborhood watch reduce crime? A systematic review and meta-analysis," *Journal of Experimental Criminology,* vol. 2, pp. 437-458, 2006.

[41] V. Grover, R. Adderley, and M. Bramer, "Review of current crime prediction techniques," in *Applications and Innovations in Intelligent Systems XIV*, ed: Springer, 2007, pp. 233-237.

[42] H. Banaee, M. U. Ahmed, and A. Loutfi, "Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges," *Sensors,* vol. 13, pp. 17472-17500, 2013.

[43] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *Communications Surveys & Tutorials, IEEE,* vol. 16, pp. 414-454, 2014.

[44] (2015). *IoT-A*. Available: http://www.iot-a.eu/public

[45] (2015). *FIRE*. Available: http://www.ict-fire.eu/home.html

[46] Y. Jeen, J. Park, and P. Park, "Design and implementation of the smart healthcare frame based on pervasive computing technology," in *Advanced Communication Technology, The 9th International Conference on*, 2007, pp. 349-352.

[47] A. Ghosh, Y.-K. Hui, and M. Chiang, "Model-based architecture analysis for wireless healthcare," in *Proceedings of the First ACM MobiHoc Workshop on Pervasive Wireless Healthcare*, 2011, p. 12.

[48] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, *et al.*, "An advanced wireless sensor network for health monitoring," in *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, 2006, pp. 2-4.

[49] U. Mitra, B. A. Emken, S. Lee, M. Li, V. Rozgic, G. Thatte, *et al.*, "KNOWME: a case study in wireless body area sensor network design," *Communications Magazine, IEEE,* vol. 50, pp. 116-125, 2012.

[50] M. R. Yuce, "Implementation of wireless body area networks for healthcare systems," *Sensors and Actuators A: Physical,* vol. 162, pp. 116-129, 2010.

[51] C. McGregor, "A cloud computing framework for real-time rural and remote service of critical care," in *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, 2011, pp. 1-6.

[52] I. Gronbaek, "Architecture for the Internet of Things (IoT): API and interconnect," in *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, 2008, pp. 802-807.

[53] G. Dai and Y. Wang, "Design on architecture of internet of things," in *Advances in Computer Science and Information Engineering*, ed: Springer, 2012, pp. 1-7.

[54] L. Tan and N. Wang, "Future internet: The internet of things," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, 2010, pp. V5-376-V5-380.

[55] M. Kovatsch, S. Mayer, and B. Ostermaier, "Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, 2012, pp. 751-756.

[56] H. Ning and Z. Wang, "Future Internet of things architecture: like mankind neural system or social organization framework?," *Communications Letters, IEEE,* vol. 15, pp. 461-463, 2011.

[57] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *Services Computing, IEEE Transactions on,* vol. 3, pp. 223-235, 2010.

[58] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, *et al.*, "SOA-based integration of the internet of things in enterprise services," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009, pp. 968-975.

[59] T. Erl, "Service-oriented architecture (SOA): concepts, technology, and design," 2005.

[60] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "Role of middleware for internet of things: A study," *International Journal of Computer Science & Engineering Survey (IJCSES),* vol. 2, pp. 94-105, 2011.

[61] C. Associati, "The evolution of internet of things," *Focus. Milão, fev,* 2011.

[62] A. Bassi and G. Horn, "Internet of Things in 2020: A Roadmap for the Future," *European Commission: Information Society and Media,* 2008.

[63] C. Huo, *A Centralized IoT Middleware System for Devices Working Across Application Domains Using Self-descriptive Capability Profile*: University of California, Irvine, 2014.

[64] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature,* vol. 457, pp. 1012-1014, 2009.

[65] C. P. McGregor, "System, method and computer program for multi-dimensional temporal data mining," ed: Google Patents, 2013.

[66] M. Sung, C. Marci, and A. Pentland, "Journal of NeuroEngineering and Rehabilitation," *Journal of neuroengineering and rehabilitation,* vol. 2, pp. 0003-2, 2005.

[67] U. Anliker, J. Ward, P. Lukowicz, G. Tröster, F. Dolveck, M. Baer*, et al.*, "AMON: a wearable multiparameter medical monitoring and alert system," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, pp. 415-427, 2004.

[68] C. D. Katsis, G. Ganiatsas, and D. I. Fotiadis, "An integrated telemedicine platform for the assessment of affective physiological states," *Diagnostic Pathology,* vol. 1, p. 16, 2006.

[69] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, pp. 1-12, 2010.

[70] B. Rao, P. Saluia, N. Sharma, A. Mittal, and S. Sharma, "Cloud computing for Internet of Things & sensing based applications," in *Sensing Technology (ICST), 2012 Sixth International Conference on*, 2012, pp. 374-380.

[71] M. Weiser, "The computer for the 21st century," *Scientific american,* vol. 265, pp. 94-104, 1991.

[72] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE,* vol. 8, pp. 22-32, 1994.

[73] G. D. Abowd and E. D. Mynatt, "Charting past, present, and future research in ubiquitous computing," *ACM Transactions on Computer-Human Interaction (TOCHI),* vol. 7, pp. 29-58, 2000.

[74] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and ubiquitous computing*, 1999, pp. 304-307.

[75] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-computer interaction,* vol. 16, pp. 97-166, 2001.

[76] M. M. Molla and S. I. Ahamed, "A survey of middleware for sensor network and challenges," in *Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on*, 2006, pp. 6 pp.-228.

[77] K. E. Kjær, "A survey of context-aware middleware," in *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*, 2007, pp. 148-155.

[78] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing,* vol. 2, pp. 263-277, 2007.

[79] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Computing Surveys (CSUR),* vol. 44, p. 24, 2012.

[80] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan*, et al.*, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing,* vol. 6, pp. 161-180, 2010.

[81] A. Saeed and T. Waheed, "An extensive survey of context-aware middleware architectures," in *Electro/Information Technology (EIT), 2010 IEEE International Conference on*, 2010, pp. 1-6.

[82] K. B. Balavalad, S. Manvi, and A. Sutagundar, "Context aware computing in wireless sensor networks," in *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on*, 2009, pp. 514-516.

[83] A. Bikakis, T. Patkos, G. Antoniou, and D. Plexousakis, "A survey of semantics-based approaches for context reasoning in ambient intelligence," in *Constructing ambient intelligence*, ed: Springer, 2008, pp. 14-23.

[84] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys (CSUR),* vol. 35, pp. 114-131, 2003.

[85] R. Baldoni, M. Contenti, S. T. Piergiovanni, and A. Virgillito, "Modeling publish/subscribe communication systems: towards a formal approach," in *Object-Oriented Real-Time Dependable Systems, 2003.(WORDS 2003). Proceedings of the Eighth International Workshop on*, 2003, pp. 304-311.

[86] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery,* vol. 24, pp. 515-554, 2012.

[87] B. Guo, Z. Yu, D. Zhang, and X. Zhou, "Cross-community sensing and mining," *Communications Magazine, IEEE,* vol. 52, pp. 144-152, 2014.

[88] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *Communications Magazine, IEEE,* vol. 49, pp. 32-39, 2011.

[89] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu*, et al.*, "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, 2006, pp. 125-138.

[90] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of things," *Internet of Things Journal, IEEE,* vol. 1, pp. 112-121, 2014.

[91] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* vol. 370, pp. 176-197, 2012.

[92] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet* vol. 43: John Wiley & Sons, 2011.

[93] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks,* vol. 56, pp. 3594-3608, 2012.

[94] S. K. Datta, C. Bonnet, and N. Nikaein, "An IoT gateway centric architecture to provide novel M2M services," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, 2014, pp. 514-519.

[95] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*, ed: Springer, 2018, pp. 103-130.

[96] F. Baccelli, N. Khude, R. Laroia, J. Li, T. Richardson, S. Shakkottai, *et al.*, "On the design of device-to-device autonomous discovery," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, 2012, pp. 1-9.

[97] G. Sarosi, J. Thomas, W. Helms, and C. Cholas, "Method and system for device discovery and content management on a network," ed: Google Patents, 2016.

[98] P. C. Ccori, L. C. C. De Biase, M. K. Zuffo, and F. S. C. da Silva, "Device discovery strategies for the IoT," in *Consumer Electronics (ISCE), 2016 IEEE International Symposium on*, 2016, pp. 97-98.

[99] Q. M. Ashraf, M. H. Habaebi, M. R. Islam, and S. Khan, "Device discovery and configuration scheme for Internet of Things," in *Intelligent Systems Engineering (ICISE), 2016 International Conference on*, 2016, pp. 38-43.

[100] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *ACM SIGCOMM Computer Communication Review*, 2009, pp. 267-278.

[101] D.-T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Computer & Information Sciences,* vol. 9, pp. 219-242, 1980.

[102] J. An, X. Gui, Z. Wang, J. Yang, and X. He, "A crowdsourcing assignment model based on mobile crowd sensing in the internet of things," *IEEE Internet of Things Journal,* vol. 2, pp. 358-369, 2015.

[103] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 173-184.

[104] X. Zhang, Z. Yang, C. Wu, W. Sun, Y. Liu, and K. Xing, "Robust trajectory estimation for crowdsourcing-based mobile applications," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, pp. 1876-1885, 2014.

[105] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 50-61.

[106] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine,* vol. 49, 2011.

[107] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal,* vol. 4, pp. 1-20, 2017.

[108] C. C. Aggarwal and T. Abdelzaher, "Social sensing," in *Managing and mining sensor data*, ed: Springer, 2013, pp. 237-297.

[109] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," *Computer Science and Information Systems,* vol. 8, pp. 1207-1228, 2011.

[110] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its application to trust-based routing," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, pp. 1732-1738.

[111] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports,* vol. 659, pp. 1-44, 2016.

[112] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, "Network analysis in the social sciences," *science,* vol. 323, pp. 892-895, 2009.

[113] J. A. Bondy and U. S. R. Murty, *Graph theory with applications* vol. 290: Macmillan London, 1976.

[114] S. Fortunato, "Community detection in graphs," *Physics reports,* vol. 486, pp. 75-174, 2010.

[115] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics,* vol. 6, pp. 29-123, 2009.

[116] *Model-based Approach to Detecting Densely Overlapping Communities in Networks*. Available: http://snap.stanford.edu/agm/

[117] J. Yang and J. Leskovec, "Community-affiliation graph model for overlapping network community detection," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 2012, pp. 1170-1175.

[118] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems,* vol. 7, pp. 358-386, 2005.

[119] P. Franti, O. Virmajoki, and V. Hautamaki, "Fast agglomerative clustering using a k-nearest neighbor graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1875-1881, 2006.

[120] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment,* vol. 2008, p. P10008, 2008.

[121] E. Le Martelot and C. Hankin, "Multi-scale Community Detection using Stability Optimisation within Greedy Algorithms," *arXiv preprint arXiv:1201.3307,* 2012.

[122] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences,* vol. 103, pp. 8577-8582, 2006.

[123] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences,* vol. 105, pp. 1118-1123, 2008.

[124] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Pacific symposium on biocomputing*, 2001, pp. 6-17.

[125] D. J. MacKay, *Information theory, inference and learning algorithms*: Cambridge university press, 2003.

[126] R. Jalali, S. Koohi-Fayegh, K. El-Khatib, D. Hoornweg, and H. Li, "Investigating the potential of ridesharing to reduce vehicle emissions," *Urban Planning,* vol. 2, p. 26, 2017.

[127] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences,* vol. 111, pp. 13290-13294, 2014.

[128] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research,* vol. 223, pp. 295-303, 2012.

[129] W. He, K. Hwang, and D. Li, "Intelligent carpool routing for urban ridesharing by mining GPS trajectories," *IEEE Transactions on Intelligent Transportation Systems,* vol. 15, pp. 2286-2296, 2014.

[130] T. Fujisaka, R. Lee, and K. Sumiya, "Exploring urban characteristics using movement history of mass mobile microbloggers," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, 2010, pp. 13-18.

[131] A. Noulas, C. Mascolo, and E. Frias-Martinez, "Exploiting foursquare and cellular data to infer user activity in urban environments," in *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, 2013, pp. 167-176.

[132] S. Wakamiya, R. Lee, and K. Sumiya, "Urban area characterization based on semantics of crowd activities in twitter," *GeoSpatial Semantics,* pp. 108-123, 2011.

[133] B. Cici, A. Markopoulou, E. Frias-Martinez, and N. Laoutaris, "Assessing the potential of ride-sharing using mobile and social data: a tale of four cities," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 201-211.

[134] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 27, pp. 13-18, 1979.

[135] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics),* vol. 28, pp. 100-108, 1979.

[136] M. M. Deza and E. Deza, "Encyclopedia of distances," in *Encyclopedia of Distances*, ed: Springer, 2009, pp. 1-583.