Estimating Bus Passengers' Origin-Destination of Travel Route Using Data Analytics on Wi-Fi and Bluetooth Signals

Shahrzad Jalali

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the Master of Science degree in Electronic Business Technologies

Supervisor: Dr. Bijan Raahemi

School of Electrical Engineering and Computer Science (EECS)

Faculty of Engineering

University of Ottawa

ACKNOWLEDGMENT

It would not have been possible to work on this research without the help and support of many people, and it is my pleasure to thank all of them.

First of all, I would like to express my sincere gratitude to my supervisor Prof. Bijan Raahemi for the continuous support, motivation, and immense knowledge that have made my research possible.

I sincerely thank my dear colleagues in this project Dr. Amir H. Ghods and Dr. Hamed H. Afshari for their contributions, and cooperation.

I would also appreciate SMATS Traffic Solutions company, Ontario Centre of Excellence (OCE) and NSERC for their financial support.

Finally, I extend my sincerest thanks to my dear mother and my kind husband, Amir Reza, for their love and support. My family has always encouraged me throughout my studies and more importantly my life.

ABSTRACT

Accurate estimation of Origin and Destination (O-D) of passengers has been an essential objective for public transit agencies because knowledge of passengers' flow enables them to forecast ridership, and plan for bus schedules, and bus routes. However, obtaining O-D information using traditional ways, such as conducting surveys, cannot fulfill today's requirements of intelligent transportation and route planning in smart cities.

Estimating bus passengers' O-D using Wi-Fi and Bluetooth signals detected from their mobile devices is the primary objective of this project. For this purpose, we collected anonymized passengers' data using SMATS TrafficBox^{TM} sensor provided by "SMATS Traffic Solutions" company. We then performed pre-processing steps including data cleaning, feature extraction, and data normalization, then, built various models using data mining techniques. The main challenge in this project was to distinguish between passengers' and non-passengers' signals since the sensor captures all signals in its surrounding environment including substantial noise from devices outside of the bus. To address this challenge, we applied Hierarchical and K-Means clustering algorithms to separate passengers from non-passengers' signals automatically. By assigning GPS data to passengers' signals, we could find commuters' O-D. Moreover, we developed a second method based on an online analysis of sequential data, where specific thresholds were set to recognize passengers' signals in real time. This method could create the O-D matrix online.

Finally, in the validation phase, we compared the ground truth data with both estimated O-D matrices in both approaches and calculated their accuracy. Based on the final results, our proposed approaches can detect more than 20% of passengers (compared to 5% detection rate of traditional survey-based methods), and estimate the origin and destination of passengers with an accuracy of about 93%.

With such promising results, these approaches are suitable alternatives for traditional and time-consuming ways of obtaining O-D data. This enables public transit companies to enhance their service offering by efficiently planning and scheduling the bus routes, improving ride comfort, and lowering operating costs of urban transportation.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

ACK	Acknowledgment
AFC	Automatic Fare Collection
APC	Automatic Passenger Counter
AVL	Automatic Vehicle Location
CRISP	Cross-Industry Standard Process
DSR	Design Science Research
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IPTS	Intelligent Public Transportation Systems
ITS	Intelligent Transportation System
MAC	Media Access Control
NIC	Network Interface Controller
O-D	Origin-Destination
OSM	OpenStreetMap
OSRM	Open Source Routing Machine
OUI	Organizationally Unique Identifier
RSSI	Received Signal Strength Indication

Chapter 1: Introduction

Traffic congestion is one of the major problems in big cities, causing delays in daily commutes, resulting in financial loss and time waste, and negatively impacting the environment and quality of life. Public transportation systems, offering high capacities and low environmental pollution, are considered a suitable solution for efficient use of the roads and reducing traffic jams in the cities. However, to encourage frequent use of this transit mode, it must be reliable, well-designed and comfortable.

Intelligent Transportation System (ITS) utilizes multi-disciplinary technologies to enable analyzing and exploiting existing facilities in ground transit to enhance its reliability, comfort, and efficiency (Elkosantini & Darmoul, 2013). The primary goal of ITS is to solve traffic problems by sensing, analyzing and controlling related operational data. One of ITS application is in advanced public transportation systems with the aim of boosting its effectiveness (Choudhary, 2018).

Among different factors for the design and efficient operation of the public transportation system, transit commuters' Origin-Destination (O-D) data is an essential requirement (Ji, Zhao, Zhang, & Du, 2017). O-D data indicate the endpoints of each trip and provides the ridership patterns of commuters. In other words, O-D shows where each passenger's trip started and where it is ended. However, obtaining origin and destination data is problematic and expensive in general, especially in large cities with extensive public transport routes.

ITS can assist in collecting O-D data from commuters. In addition to O-D onboard surveys, different data sources such as Automatic Fare Collection (AFC), Automatic Passenger Counter (APC), and Automatic Vehicle Location (AVL) can be used as inputs in ITS systems. However, due to the shortcomings of each of these data sources, there is a need for new ways of estimating the O-D matrix. In recent studies, Wi-Fi and Bluetooth sensors are employed for collecting O-D data in addition to the above datasets.

1.1. Research Motivation

Based on the report from Statistics Canada, Canadian employees are using public transit more than before, and the number of public transportation passengers has increased by 59.5% from 1996 to 2016 (Statistics Canada, 2017). However, it has also been reported that the average time public transit commuters spent to reach their workplaces, which included both waiting times in stations and duration of trips increased by 1.9 minutes, becoming 44.8 minutes in 2016 (Statistics Canada, 2017). Therefore, it is necessary to regularly enhance bus scheduling and routes design in order to reduce wait time and make public transportation more efficient and convenient.

Estimating Origin-Destination (O-D) of public transit passengers offers information about their ridership patterns. Also, it can specify which station is crowded during different periods of a day, month or season. This information assists public transit agencies in making strategic decisions and effectively planning and scheduling bus routes. Furthermore, it can reduce environmental negative impacts and cost of fuel consumption by efficient use of buses capacity. However, as obtaining O-D information is non-trivial and costly, we propose new approaches based on sensing technology to estimate O-D information automatically.

In our proposed approaches, we employ Wi-Fi and Bluetooth technologies to anonymously capture the MAC addresses of bus passengers' portable devices such as cell phones, wireless headsets, and smart watches and then correlating this data with bus GPS locations. By applying data mining techniques, in particular, clustering and time series sequential analysis, to bus passengers' dataset, we can estimate their origins and destinations with reasonable accuracy.

1.2. Research Problem

Although O-D information is important for public transportation agencies to improve their bus network operation with bus scheduling and route design, traditional ways of gathering this data, such as onboard surveys and human observation, are laborious and costly, and it could be sensitive to human's mistake (Dunlap, Li, Henrickson, & Wang, 2016). Also, electronic ticketing devices can help to estimate the O-D matrix for public

transits such as most subways where passengers require to use their electronic tickets both for boarding and alighting. However, using automatic fare collection technologies on buses can help to find the origin data, not the destination because passengers use their e-tickets only for boarding the bus. Therefore, passengers' O-D flows should be deduced from travel behavior assumptions. However, validating these assumptions are still challenging (Mishalani, McCord, & Reinhold, 2016).

An alternative approach is to employ Automatic Passengers Count (APC) devices which count the number of passengers getting on and off at each bus stop (Kostakos, Camacho, & Mantero, 2010), then infer this data to find boarding and alighting location of passengers. Since O-D data can be obtained from APC devices indirectly, this issue causes unavoidable estimation errors. Thus, we aim to design novel methods to estimate both the origin and destination of passengers more accurately.

With the growing number of mobile devices such as smart cell phones, tablets, and laptops, we decide to employ sensing technologies for detecting passengers' mobile device signals, then estimating the O-D matrix based on these signals. In this research, we use SMATS Wi-Fi and Bluetooth sensor to collect and analyze bus passengers O-D data anonymously. These sensors are based on detecting passengers' devices Wi-Fi and Bluetooth MAC address as well as their location for obtaining O-D data. However, the primary challenge in using Wi-Fi and Bluetooth sensors is that they collect not only Wi-Fi and Bluetooth signals of passengers' device but also all the MAC addresses of other devices that are not necessarily inside the bus but are within the range of the sensors' covered signal area. Therefore, the sensors might receive and capture signals which are transmitted by mobile devices in other vehicles, pedestrian, and nearby buildings. Figure 1.1 illustrates this scenario.

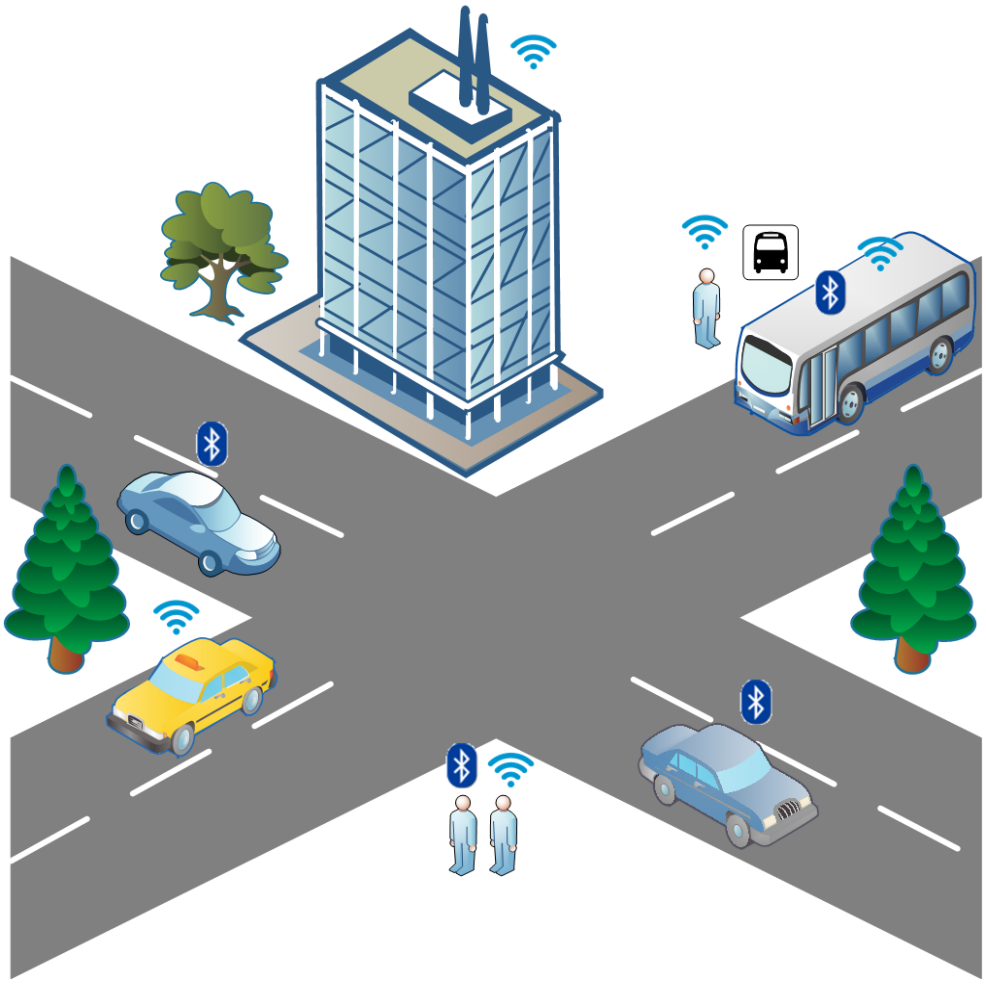

Figure 1.1 Various Wi-Fi and Bluetooth signals from the surrounding environment

As shown in Figure 1.1, distinguishing between passengers and non-passengers' signals is not a trivial task. Various cleaning and filtering methods are needed. However, applying filtering methods might eliminate the passengers' signals as well. For this reason, choosing proper filtering approaches is required to make the solution effective and general.

Table 1.1 shows the research's objectives and problem statement.

Table 1.1 Framing the research problem

Steps	Description
Observation	- Bus Passengers' Origin and Destination data can provide public transit agencies considerable insight on ridership pattern and travel choices of commuters. - O-D data can be used for understanding travel demands, route designing, and also bus scheduling. - Gaining O-D data is laborious and expensive. Therefore, the O-D survey has been held every few years.
Thesis	Estimating bus passengers' origin and destination using their mobile device signals can facilitate collecting valuable data about commuters' trip patterns. However, using sensing technologies in this application is challenging because the Wi-Fi and Bluetooth sensor can collect all the signals in the surrounding environment (i.e., noise) in addition to passengers' signals.
Enthymeme	Since traditional ways of gathering O-D data such as onboard surveys and human observation are costly, and it could be sensitive to human's mistake (Dunlap et al., 2016), proposing a new and automated approach to facilitate this process is required. The proposed methods should be able to estimate passengers' O-D accurately, and helping with route planning.
Problem Statement	Ridership patterns of passengers can change during months and years. Consequently, O-D surveys should be done regularly. However, because of its challenges and difficulties, transit agencies might tend to postpone it as long as possible. One of the main problems of O-D surveys is the number of people who can participate in them. Since a limited number of surveyors conducts the survey, they only can collect data from a limited number of respondents, as well. Wi-Fi and Bluetooth sensors do not have this limitation and can obtain a large amount of O-D data in a short time.
Objective	- Developing two different methods to facilitate the process of finding

	O-D of passengers using Wi-Fi and Bluetooth signals of their mobile devices. One of these methods applies clustering algorithms, Hierarchical agglomerative and K-means, to divide passengers from non-passengers' signals. The second method uses strict thresholds to find passengers' signals and calculates the O-D matrix in real-time. - Developing an algorithm to find the sensor's GPS data which are out of the route path and correct them. - Extracting features from time-series signals to cluster data based on them. - Comparing the two proposed methods based on their accuracy and time complexity.
Research Questions	- Is it feasible to capture passengers' device Wi-Fi and Bluetooth signals to calculate their O-D? - Can the data analytics methods distinguish between passengers and non-passengers signals properly using extracting features? - What are the accuracy and complexity of using the Wi-Fi and Bluetooth sensors? - Which approach is more accurate in estimating passengers' O-D: clustering technique, or threshold-based signals analysis?

1.3. Thesis Objectives

In this research, we aim to estimate the origin-destination data of bus passengers using their devices' Wi-Fi and Bluetooth signals. For this purpose, we collect passengers' devices Wi-Fi and Bluetooth MAC addresses, their signal strength, as well as the location of the bus (GPS signal), in order to accurately estimate where the passengers get on and off the bus. We apply Clustering methods, as well as sequential event analysis to correlate events such as bus arrival and departure times with time stamps of Bluetooth and Wi-Fi MAC addresses captured by the sensors. This analysis will help us to remove noise from the dataset.

The real data will be collected by sensors from SMATS Traffic Solutions, our industry partner in this research, and will be anonymized for privacy protection. The anonymized data will be at disposal for further analysis. The solutions we propose in this research improve the efficiency of public transportation systems by facilitating efficient bus scheduling and route planning.

1.4. Methodology

In this study, Design Science Research (DSR) methodology is used to develop two different methods to achieve the objectives of the research. DSR methodology creates and evaluates the solutions to meet business requirements and solve organizational problems (Hevner, March, Park, & Ram, 2004). Accordingly, we aligned our methods with the DSR guidelines.

According to Peffers et al. (2007), Design Science Research methodology consists of six stages. The first step is identifying problems and research motivation. In this stage, the research problems should be defined, and the research motivation should be specified. The second step is defining objectives of a solution which can be either quantitative or qualitative. Based on the definition of the problems, these objectives are specified. The next step is designing and developing the artifact. The form of the artifact depends on the research contribution type, and it can be either a designed theory, a model, a method, or a software product. The fourth step is demonstrating, in which the artifact should be used to solve the research problems. Before that, the proper context for using the artifact should be found. The fifth step is evaluating and measuring how effective and efficient the artifact is in solving the research problems. For this purpose, the objectives of the solution can be compared with the artifact results from step four. After this step, the researcher might iterate back to the design step in order to modify and improve the artifact design. Finally, the sixth step is communicating the final results regarding scholarly and professional publications. The steps in the DSR methodology are shown in Figure 1.2.

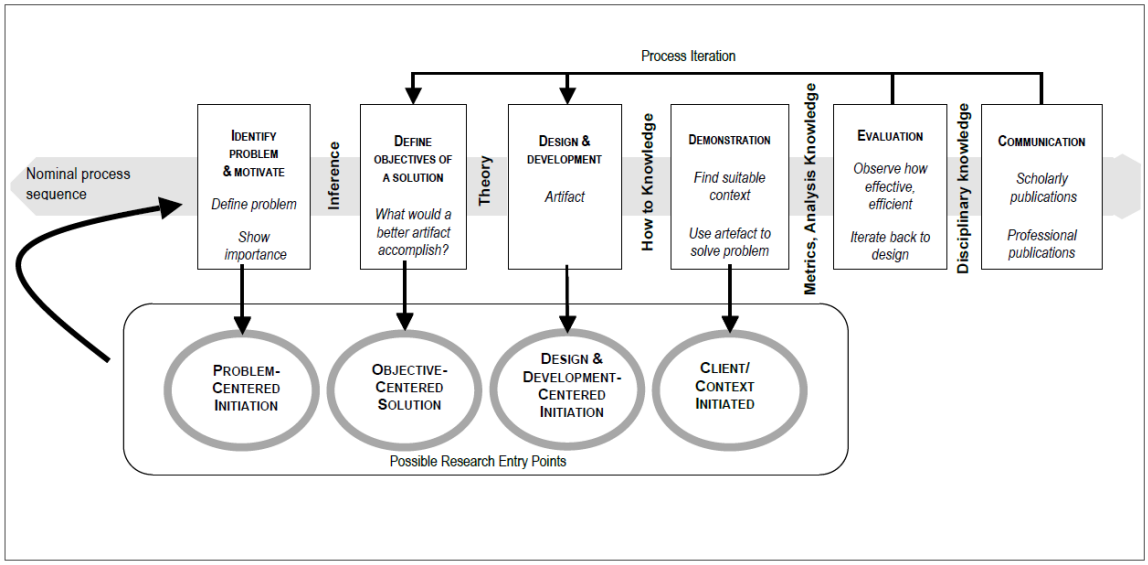

Figure 1.2 The Design Science Research Methodology (Peffers et al., 2007)

Following the DSR methodology, we have performed different activities throughout its six different steps. In the first stage, problem identification, because we have been collaborating with SMATS Traffic Solutions company since the beginning of this research, we could understand their business needs and identify their motivation in a timely fashion.

In the second phase, our primary objective is to develop a novel and optimal method for estimating bus passengers' Origin and Destination using the SMATS TrafficBox ™ sensor. The SMATS TrafficBox ™ sensor consists of both Wi-Fi and Bluetooth modules detect devices' Wi-Fi and Bluetooth MAC addresses from their transmitted signals. By studying the literature, we learn about the recent state of the art methods which have been proposed for O-D estimation.

In the design and development phase, we designed and developed two models in python (using Pandas, Numpy, Sklearn, and Matplotlib libraries) to identify passengers from non-passengers' signals. The first method applies clustering algorithms to solve the O-D estimation problem, whereas the second approach uses threshold-based analysis of sequential events to estimate alighting and boarding location of passengers.

The first model uses clustering algorithms (K-Means and Hierarchical Agglomerative methods) to divide passengers' data from non-passengers based on the

extracted features. Also, the GPS data is used to find passengers' Origin-destination based on the time of their first and last detected signals. Therefore, we can easily estimate the Origin-Destination of passengers' cluster(s).

The second model is an online threshold-based approach which analyzes the data in real-time. Since the data is analyzed online, we set some threshold to remove signals which do not belong to passengers. Also, the frequency of sending signals for each Wi-Fi and Bluetooth device is related to several factors, and it is possible that a device is onboard but not sending probe request for a while. Therefore, we considered another threshold to check if the passengers are still onboard or not. We define that, if SMATS TrafficBox™ sensor could not detect a device for three consecutive stops, this passenger was alighted three bus stations before.

In the demonstration phase, the first step is data collection. For gathering data, we placed the TrafficBox™ sensor inside a 40-feet bus for six days from 4:00 PM until 5:30 PM (note: the ethics approval was secured before collecting the data anonymously. Please see Appendix A for a copy of the Ethics Certificate). During this time, the sensor captured not only transmitted Wi-Fi and Bluetooth MAC addresses from mobile devices also the sensor's GPS data, which were then stored in its memory unit. Since the sensor detects all the signals in the surrounding environment, the collected signals were not just for passengers' devices. They include noise and unwanted signals as well. Therefore, we had to find a solution to filter passengers from non-passengers' signals. Figure 1.3 demonstrates the data collection process.

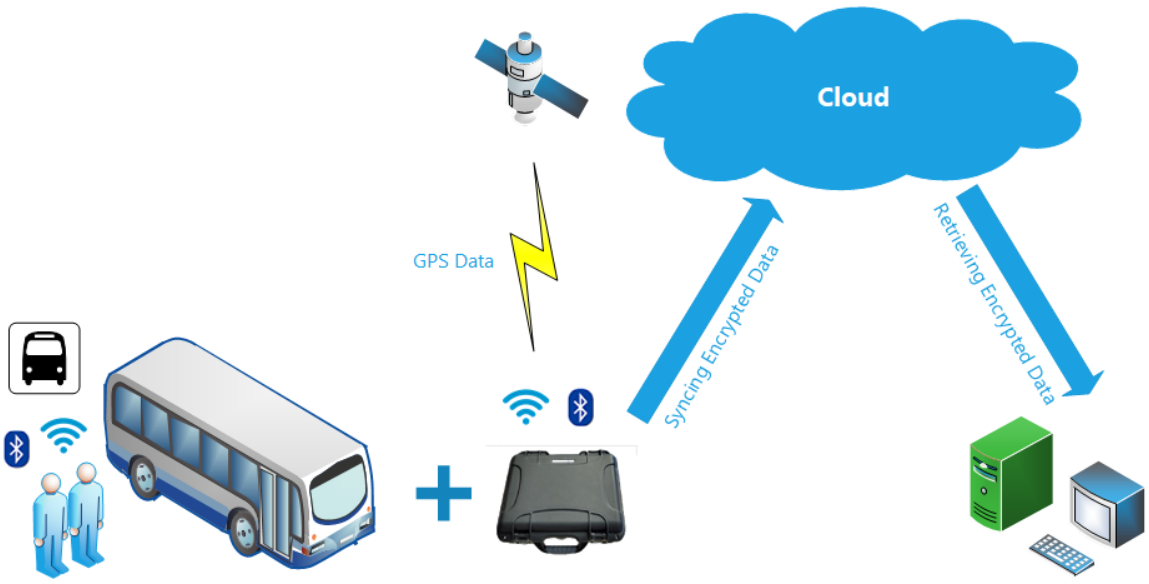

Figure 1.3 The data collection process

Our first developed method employs a clustering solution to separate the signals inside and outside of the bus. For clustering purpose, we correlated signals data set with GPS data using their timestamps. As a result, we assigned corresponding Latitude and Longitude to each signal record. Then, we extracted five different features from the signals data set to use in clustering algorithms. Based on these features, clustering algorithms automatically could differentiate between passengers and non-passengers' signals. Also, we tried various clustering methods to discover which one performs well in this particular application. Finally, we estimated the passengers' O-D matrix using the GPS data.

Moreover, we proposed a second technique, namely online threshold-based analysis. This technique analyzes the passengers' device Wi-Fi and Bluetooth signals online to estimate their boarding and alighting location in real time. For this purpose, we employed some strict thresholds to filter out the passengers' signals and discard the other ones (noise, non-passengers' signals).

For the evaluation step, we used the percent error validation method to compare the ground truth results with the estimated one, and also, compare these proposed methods results with each other.

In the communication phase, we communicated the results of this research with SMATS Traffic Solutions Company, and also, our peers in the forms of conference publication and poster presentations.

1.5. Thesis Contribution

This research focuses on proposing two new methods for estimating bus passengers' origin and destination using Wi-Fi and Bluetooth sensors. These proposed methods facilitate the process of conducting O-D surveys, and also, are suitable alternatives for traditional ways of conducting O-D surveys. Using Wi-Fi and Bluetooth sensors to find commuters' O-D has been addressed in several studies (Baeta, Fernandes, & Ferreira, 2016; Dunlap et al., 2016; Fukuda et al., 2017; Ji et al., 2017). However, our contribution is using clustering methods for automatically detecting non-passengers' signals from the data set and eliminating them. Previous studies established strict thresholds manually which makes their solution not automated and general. Also, to the best of our knowledge, there is no study about online estimating O-D of passengers which makes it possible for public transit companies to monitor and control the passengers' flows in real time.

We also develop a method for correcting GPS data especially for downtown areas where high-rise buildings are blocking the GPS signals and causing less accuracy in those areas. GPS data facilitate estimating bus passengers' boarding and alighting location. Therefore, it should be accurate as much as is possible. Since the accuracy of captured GPS data depends on several factors (including structural obstacles), sometimes it happens that the GPS data is not recorded on the traveled route. While this challenge is common for all GPS modules, no studies have already pointed out how they addressed this issue.

Moreover, we evaluate the final O-D results with ground truth data, and we calculate the percent error of our proposed methods for the six trips taken. For collecting the ground truth data, two observers were onboard during each trip and recorded passengers boarding and alighting location in addition to a number of passengers who are getting on or off at each bus stop. However, in current studies, the estimated O-D

evaluation based on the actual one is missing because recording all the passengers' flows in the real field is challenging. Accordingly, the O-D comparison versus the ground truth was not provided in the previously published literature. Indeed, they have only compared their final results with APC data to validate their results while the estimated O-D was not assessed.

In summary, the contributions of this research are:

- Automated clustering techniques for O-D estimation
- Threshold-based analysis of the sequential events for O-D estimation in real-time
- Utilizing the GPS data in the analysis of the data for O-D estimation
- Establishing ground truth evaluation for validation of the proposed techniques

1.6. Publications/Presentation Resulted from This Research

The conference publication and poster presentations that resulted from this research are as follows:

1- Afshari, H. H., Jalali, S., Ghods, A. H., & Raahemi, B. (2019). An Intelligent Traffic Management System Based on the Wi-Fi and Bluetooth Sensing and Data Clustering. In K. Arai, R. Bhatia, & S. Kapoor (Eds.), *Proceedings of the Future Technologies Conference (FTC) 2018* (pp. 298–312). Cham: Springer International Publishing.

2- Jalali, S., Afshari, H. H., Ghods, A. H., & Raahemi, B (2019). Estimation of Bus Passengers' Origin-Destination Using Clustering Method on Wi-Fi and Bluetooth Data. Extended Abstract and Poster Presentation in *Transportation Research Board 2019 Annual Meeting, USA.*

3- Jalali, S. & Raahemi (2018). Estimating Bus Passengers' Origin-Destination Travel Route using Data Analytics on Wi-Fi and Bluetooth Signals. Poster Presentation in *Engineering and computer science Graduate Poster competition, University of Ottawa*.

4- Jalali, S. & Raahemi (2017) Estimating Bus Passengers' Origin-Destination Travel Route using Data Analytics on Wi-Fi and Bluetooth Signals. Oral

Presentation and Poster Presentation in *National Capital Region Thesis Competition and Poster Session (Sprott-Telfer-UQO).*

1.7. Thesis Outline

The thesis is organized in six chapters. In Chapter 2, we first explain different types of Intelligent transportation systems and Intelligent public transit. Then, we review the literature on the O-D estimation and the state of the art methods for determining the boarding and alighting locations of passengers. We also briefly explain Wi-Fi and Bluetooth technologies, and their features.

In Chapter 3, we explain our first proposed method which is based on the clustering solution. We describe the different experiments we did and explain different stages in data preprocessing which includes data cleaning, GPS data modification, GPS assignments to signal records, feature extraction, and data normalization. Also, we apply two different clustering methods, Hierarchical agglomerative and K-Means, to identify passengers' clusters, and report on the results.

In Chapter 4, we propose our second approach which analyzes the sensor data in real time. In this chapter, we explain how this approach can recognize passengers' signals and estimate their O-D online by employing thresholds, and report the final results.

In Chapter 5, we use the results of clustering and online threshold-based analysis we performed in chapter 3 and 4, respectively, to create estimated O-D matrices for all six trips taken by bus. Since we have the ground truth data, we will compare and evaluate the results of our proposed methods with the actual ones. Then, we report on the accuracy of the proposed methods and identify which method works well for this particular application.

In Chapter 6, we conclude this research by providing a summary of our research contributions, business implications, limitations, and future works.

Chapter 2: Background Study and Related Works

In this chapter, we first cover a brief background on intelligent transportation systems, intelligent public transit, and the Wi-Fi /Bluetooth technologies. We then review the state of the art literature on the Origin-Destination estimation for determining the boarding and alighting locations of passengers.

2.1. Background Study

2.1.1. Intelligent Transportation Systems

The transportation industry is progressing rapidly in the world thanks to technologies such as sensing, communications, and data analysis that helps the transportation sector to improve and become more advanced. For this reason, intelligent transportation systems have emerged and according to the Intelligent Transportation Systems Society of Canada, ITS can be defined as "The application of advanced and emerging technologies such as computers, sensors, control, communications, and electronic devices in transportation to save lives, time, money, energy, and the environment" (ITSCanada, n.d.). In other words, ITS analyzes traffic information such as travel time and velocity with the aim of reducing traffic congestion and managing transit infrastructure (Elkosantini & Darmoul, 2013). ITS includes six main applications which are as follows (Zhang et al., 2011):

1. *Advanced traffic management systems*: These systems are designed to control and manage traffic flows in real-time using data from cameras, speed sensors and so on.
2. *Advanced traveler information systems*: These systems support and assist travelers on their ways from an origin location to destination by providing useful information about the best route, traffic jam, collision, and alternative paths.
3. *Advanced vehicle control systems*: These systems have been designed with the aim of facilitating driving and enhancing road safety. The autonomous vehicle is an instance of these systems.

4. *Fleet management*: Managing and controlling all the vehicles which are used for the business purpose is called fleet management. Fleet management helps companies in managing vehicle's drivers, speed, location, and fuel consumption as well as their maintenance and costs.
5. *Advanced public transportation systems*: These systems aim to improve the efficiency and comfort of public transit mode and increase the safety of passengers using sensing and information technologies.
6. *Advanced rural transportation systems*: These systems are recently developed and employed as an application of ITS. Rural transportation systems manage transits within and through rural areas.

Each of these applications employs different kinds of technologies to facilitate transit operation.

2.1.2. Intelligent Public Transportation Systems

The main application of Intelligent Public Transportation Systems (IPTS) is gathering and providing useful information regarding conditions of transport networks (Elkosantini & Darmoul, 2013). In other words, IPTS uses different sources of operational data to analyze traffic flow and passengers' behavior and employs this information to enhance the performance and quality of public transit services.

There are different types of IPTS system which assist the improvement of public transit such as Automatic Vehicle Location system (AVL), online passenger information, Automatic Passenger Counter (APC), and Automatic Fare Collection (AFC). APC uses infrared technology to count the number of passengers who are alighting and boarding at each bus stops. This electronic device is installed above the doorway of buses so it can provide real-time data about passengers' flow. AVL systems can record a public vehicle location. As a result, it is possible to track the vehicle as well as monitoring its speed and route (Elkosantini & Darmoul, 2013). Also, AVL data can be employed to inform passengers about the arriving time and location of buses in real time.

Automated fare collection systems are used to collect fares from riders' smart cards. Indeed, they are the replacement of manual ticketing systems which ride fare was

paid by cash or paper tickets. Currently, passengers are only required to tap their smart cards on AFC systems. For public transit modes such as subways, passengers have to use their cards both at entry and exit locations because the fare is variable and depends on the distance that commuters have traveled. Thus, in this transit mode, AFC systems can provide both alighting and boarding locations of each passenger and their related time stamps which make obtaining O-D matrix easy. However, these systems cannot provide the alighting location of passengers for bus riders as the fare is fixed and there is no need for the passengers to tap smart cards on APC systems anymore.

2.1.3. Origin-Destination Matrix

The essential requirement for designing efficient public transportation systems is collecting data related to the passengers' movements, which is called the origin-destination matrix. This matrix is essentially a table which shows the movement of passengers between origin zones and destination zones on the map (Kostakos et al., 2010). OD matrix allows estimation of the movement patterns of passengers and future demand on the transportation systems which ultimately improve bus network operation regarding bus scheduling and route planning.

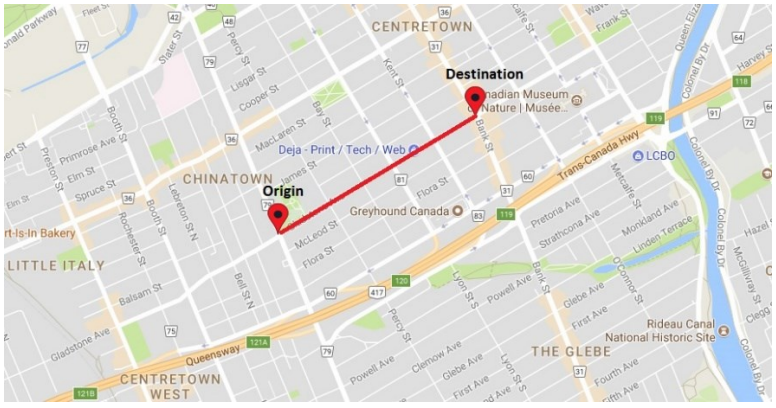

Figure 2.1 A sample Origin-Destination data on Google map

Table 2.1 shows a sample O-D matrix for a route with four bus stations. As it is shown, the O-D matrix is a square matrix, and the values on the main diagonal are all zero since it is not possible that the origin and destination of a passenger be the same. Also, the

values above and below the main diagonal indicate the number of passengers in this round-trip route.

Table 2.1 A sample Origin-Destination Matrix

	Number of Rides by Destination			
	Stop 1	***Stop 2***	***Stop 3***	***Stop 4***
Stop 1	0	2	4	6
Stop 2	6	0	1	7
Stop 3	3	2	0	5
Stop 4	8	5	1	0

Determining the OD matrix based on the traditional way is so expensive and problematic since it depends on surveys or human observers who count the number of passengers during a period. Also, other types of transit data such as APC, AFC, and AVL have their deficiencies. For instance, electronic ticketing systems can help to estimate the O-D matrix for public transits such as most subways where passengers require to use their electronic tickets both for boarding and alighting, but using this system on buses can help only to find the origin data of passengers without having any data about their destinations. Therefore, in this case, public transit agencies have to collect the data manually or using other sensors (APC sensors) which can count the number of passengers on a bus and cannot collect any data about O-D data of commuters (Kostakos et al., 2010).

Nowadays, researchers try to collect OD data differently by using Wi-Fi and Bluetooth sensors (Blogg, Semler, Hingorani, & Troutbeck, 2010; Dunlap et al., 2016; Ji et al., 2017; Malinovskiy, Saunier, & Wang, 2012). These sensors detect Wi-Fi and Bluetooth signals MAC addresses and use these unique addresses to differentiate passengers from non-passengers. Using the MAC addresses of devices as well as Global Positioning System (GPS) data can help to find the origin point and destination point for any passengers. The following sections investigate different O-D estimation studies and classify them according to the data sources they have used.

2.1.4. Wi-Fi Technology

Wi-Fi is a wireless networking technology which uses radio frequency to connect mobile devices to the Internet. Indeed, Wi-Fi is based on different standards which were created by the Institute of Electrical and Electronics Engineers. These standards are generally named IEEE 802.11x (Dong & Wang, 2018). Based on the standards, information is exchanged in a frame format which has several types: Beacon, Acknowledgment (ACK), Data, and Probe. Access points advertise their presence by emitting Beacon frames. Data and ACK frames are generated for exchanging information between access points and connected devices, and finally, Probe frames are sent by mobile devices that are searching for access points to connect (Ji et al., 2017). In addition, based on the IEEE 802.11 standard, there are two modes of scanning namely passive and active modes (Fukuda et al., 2017). In active mode, a mobile device actively transmits probe frames to find nearby access points, but in the passive scanning mode, the device only receives beacon frames from access points and does not send out any frames (Fukuda et al., 2017).

Wi-Fi sensors are used to discover transmitted Wi-Fi signals from different devices, and they can capture all of the frames. Each frame includes information about the sender's Media Access Control (MAC) address, frame type, time stamps, and Received Signal Strength Indication (RSSI) (Ji et al., 2017).

Each MAC address is a network address which consists of 48 bits, and it is recognized as a unique identification for each mobile device in wireless networks. With 48-bit length, it is possible to create 2^{48} different MAC addresses, and this number is large enough to make it as a unique identifier for mobile devices. Moreover, the first half of each MAC address indicates manufacturer's Organizationally Unique Identifier (OUI) and the other half specifies Network Interface Controller (NIC), which manufactures should assign it to its devices (Brennan Jr et al., 2010). Figure 2.2 shows a sample binary and its equal hexadecimal MAC address.

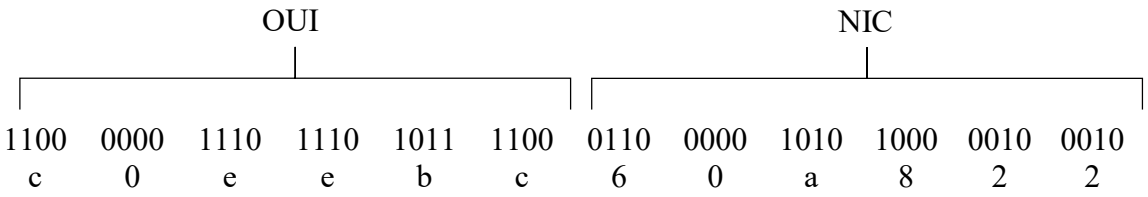

Figure 2.2 A sample MAC address

Another data which is included in Wi-Fi signals is Received Signal Strength Indication (RSSI). RSSI shows the estimation of the distance from a mobile device and a receiver device such as access points or sensors. Indeed, the stronger or greater the RSSI value, the nearer the device. However, there are several features which can affect the values of signal strength. For instance, if there were some obstacles between a mobile device and the sensor, the sensor might receive the weaker signal strength from the device due to the obstacles, but in general, this function is mostly depending on the distances (Mishalani et al., 2016)

The values of RSSI are negative and start from zero which is the best value. Its unit is in dBm (decibel per milliwatts). Figure 2.3 illustrates a sample chart of Wi-Fi RSSI values over time.

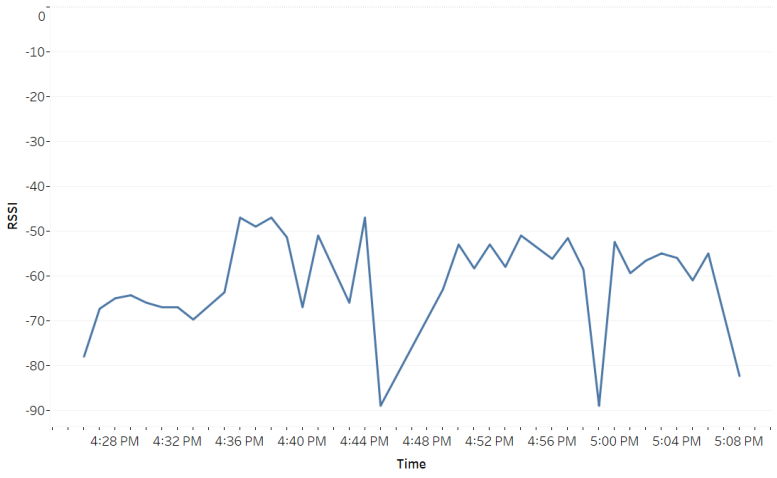

Figure 2.3 A sample chart of Wi-Fi RSSI values

2.1.5. Bluetooth Technology

Bluetooth is a wireless technology which is widely being used for short-range connectivity. A company named Ericsson developed this technology in 1994 for connecting laptops to mobile phones (Stallings, 2001). IEEE 802.15.1 is the standard for Bluetooth technology which was released by the Institute of Electrical and Electronics Engineers, and now Bluetooth Special Interest Group (SIG) is responsible for managing it. Like microwaves and Wi-Fi, this technology operates through Industrial Scientific and Medical (ISM) band at 2.4 Gigahertz (Bhaskar, Qu, Nantes, Miska, & Chung, 2015).

Bluetooth technology could connect up to eight devices together in a small network that enables them to have bi-directional communication. Also, this wireless technology has many applications in different areas. For instance, it can be used for transferring data and voice between devices, and it is a suitable replacement for cables in short-range communications (Stallings, 2001).

Bluetooth sensors are used to discover transmitted Bluetooth signals from any nearby Bluetooth devices. They are also capable of capturing any Bluetooth signals frames that are either in the discovering mode or in the paired mode. Similar to the Wi-Fi technology, each frame contains information about the sender's MAC address which acts as an identifier. Also, RSSI value can be extracted from the captured Bluetooth frames that provide a good estimation of the distance between the origin and the Bluetooth sniffer. Figure 2.4 illustrates a sample chart of Bluetooth RSSI values based on time.

Figure 2.4 A sample chart of Bluetooth RSSI values

2.2. Related Works

2.2.1. O-D Estimation with Entry-Only Boarding Locations Recorded

Boarding data can be obtained from buses' farebox because the passengers have to pay the ride fare with their electronic tickets when they get on the bus. However, finding their alighting locations is still a problem. Researchers have tried to solve this problem by using other data sources in addition to AFC data. For instance, Ji et al., (2017) combined fare box data with Wi-Fi signals to estimate both trip-level and period-level O-D flows. They claimed that their proposed Hierarchical Bayesian Model which uses both fare box and Wi-Fi datasets could estimate commuters' demand accurately. Thus, they tested their model on a bus route with 20 bus stops. Although they proposed a novel method, they did not assess the performance of the method and O-D flow matrices.

2.2.2. O-D Estimation with Neither Entry nor Exit Locations Recorded

APC data which includes passengers' number boarding and alighting at each bus stop does not provide any information about passengers' entry and exit locations. However, APC data has been used in several transit studies for O-D estimation. For example, Ji, Mishalani, & McCord (2014) employed bus APC data of several trips for estimating passengers' O-D data. For this purpose, they improved the Iterative Proportional Fitting method (IPT) and applied it to the data. The results proved that their proposed method is better than common IPT method regarding estimation O-D and computational time. Also, in another study conducted by Ji, Mishalani, & McCord (2015), APC data, as well as onboard surveys, were used to evaluate Heuristic expectation maximization method for O-D estimation.

Recently, collecting mobile device signals using Wi-Fi and Bluetooth sensors has attracted researchers' attention. These sensors are capable of capturing passengers' signals which is another source of data for O-D estimation. Although the boarding and alighting locations are not recorded in this method, locations can be assigned to the signal dataset using timestamps and sensors or buses GPS data. For instance, Dunlap et al. (2016) developed a software application which could detect nearby Wi-Fi and Bluetooth signals as well as GPS data. They performed their experiment on an urban route with

eight bus stations for four weeks. To preprocess and clean the data from noise, the authors divided Wi-Fi and Bluetooth data and also proposed several filtering methods with different thresholds for Wi-Fi and Bluetooth signals. The number of detection, trip duration, and distance from bus stops were the filtering features that they used for removing noise data. Finally, they assigned the locations to the remained signals using GPS data. Even though the authors obtained the O-D matrix; they did not evaluate the results and the accuracy of the estimated matrix. Also, they did not collect mobile devices signal strength which plays an important role in finding devices' distance from the sensor and making noise removal easier.

In Fukuda et al. (2017), the authors aimed to study the applicability of Wi-Fi signals monitoring in O-D data by performing one day test in Krabi city, Thailand. Trip duration and trip distance thresholds were used to clean Wi-Fi data. By evaluating the results with two different validation methods, they concluded that appropriate data cleaning approaches were needed to remove more outliers because their dataset still had a significant amount of noises. In another research, Mishalani, McCord, & Reinhold (2016) used Wi-Fi signals of passengers' mobile devices to determine their O-D flow. In this article, they used two filters to separate passengers' signals from the others. The first filter was for removing stationary devices' signals such as wireless routers based on the first three octets in their MAC addresses. Since these numbers are unique for each manufacturer, they can be identified and discarded easily. Also, they deleted records which their elapsed time between their first detections and last ones are more than the median cycle time. The second filter was for eliminating signals which were not considered on the bus and also, they applied distance thresholds in case of traffic congestion. After applying these filters, they created the OD flow matrix with the remaining MAC addresses. Finally, for evaluation, they used onboard surveys and Automatic Passenger Counter (APC) data.

El-Tawab et al. (2017) aimed to improve the public transit system by planning better routes and schedules for buses to increase the ridership. To achieve these objectives, they designed and developed an Intelligent system that could able to detect the majority of bus passengers at each station. This system was responsible for collecting, analyzing, and capturing passengers' MAC addresses data from all the bus stations to

cloud-based storage. First, they filtered the data for several experiments and then, they matched any single passengers in various bus stops. Moreover, Kostakos, Camacho, & Mantero (2013) and Kostakos et al. (2010) installed a Bluetooth sensor on the roof of a bus which had GPS, digital odometer, door sensors, and electronic ticketing system in Portugal to investigate passengers' behaviors. For this purpose, The researchers set a number of filters such as removing any trip duration less than 120 seconds for cleaning the dataset. Then, by assigning locations to Bluetooth records, they could estimate the O-D matrix. Ground truth data which in this research was electronic ticketing data assessed the accuracy of the estimated matrix. Also, an expert in this field confirmed their work, as well. Nonetheless, using APC data as ground truth only could evaluate the estimated O-D matrix regarding the number of detected passengers versus the recorded ones. In other words, it could not prove that if the method calculated the O-D matrix accurately.

Bai, Ireson, Mazumdar, & Ciravegna (2017) employed Wi-Fi, Bluetooth, and Bluetooth LE technologies to estimate passenger load in public transport and understand waiting times at bus stations. They collected the passengers' data in a UK metropolitan area for five days and tried different scenarios to increase accuracy. After collecting data, they used two filters to eliminate noise data. They deleted all the records with more than -90 dBm for signal strength and elapsed time less than 1 minutes. Then, they compared the ground truth data with the filtered signals in order to compute the scale factor. So the scale factor can be used with filtered signals to estimate passenger counts on the buses. However, they stated that this method could underestimate in local regions and overestimate in urban regions.

In a number of studies, Wi-Fi and Bluetooth sensors were placed in bus stops instead of buses to find the origin and destination of passengers or pedestrians. For example, Canon-Lozanol et al. (2013) proposed a method in which they installed two Bluetooth sensors in two different bus stops in Bogota, Colombia. Their primary goal was to implement an automatic web-based system that could obtain O-D Matrix using Bluetooth signals. Since they only employed two sensors which one represented the origin and the other destination, they had to eliminate records that only were detected by one sensor. Also, the authors created another filtering method for deleting signals which were not received from mobile devices based on their MAC addresses. However, they

neither evaluated the proposed method properly nor reported the accuracy clearly. Shlayan et al. (2016) presented a new system using Wi-Fi and Bluetooth technology. They suggested the encryption of the Wi-Fi and Bluetooth data which were obtained from pedestrians and then used them for real-time decision making. They ran pilot tests on two public transportation agencies in New York City and proved the usefulness of their method which helped them to understand behavioral patterns of pedestrian such as OD flows. Finally, they highlighted the importance of using location data, modified filtering techniques, sensor placement algorithms, and sensor feature modification.

Chapter 3: The Proposed Clustering Approach

In this chapter, we elaborate on our proposed approach based on clustering methods. Since this approach employs data mining algorithms as a solution for designing and demonstrating the first artifact of this research, we decided to follow the Cross-Industry Standard Process for Data Mining methodology known as CRISP-DM.

CRISP-DM methodology consists of six stages in which dependent phases are connected by arrows, and each phase includes different related tasks. The stages of the CRISP methodology include Business understanding, data understanding, data preparation, modeling, evaluation, and deployment (IBM, 2012). Figure 3.1 shows these phases and their dependencies.

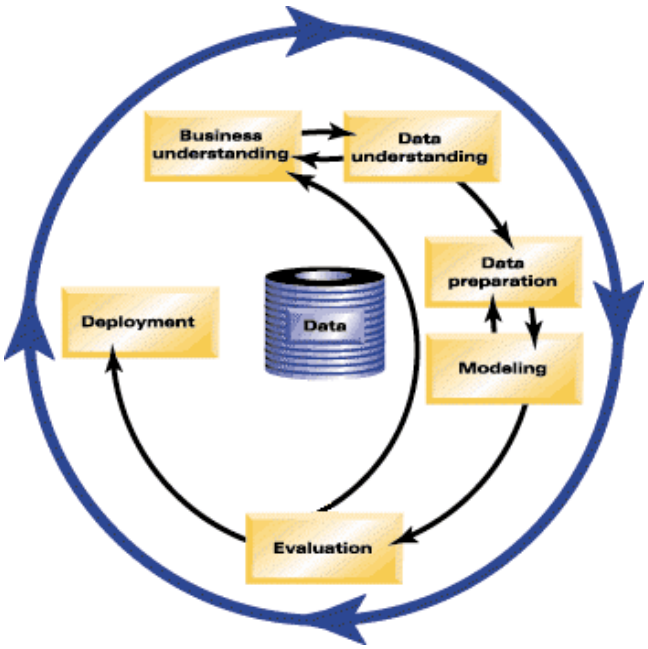

Figure 3.1 The CRISP-DM methodology process (IBM, 2012)

Following the CRISP-DM methodology of Figure 3.1, the first stage, business understanding, was addressed in Chapter 1 where we explained the motivation of this study, business needs, research questions, and the challenges, and also, elaborated on the first stage of the DSR methodology. As such, we now move on to the second stage of the CRISP-DM methodology, which is the data understanding phase. Figure 3.2 shows the overall view of the proposed clustering approach.

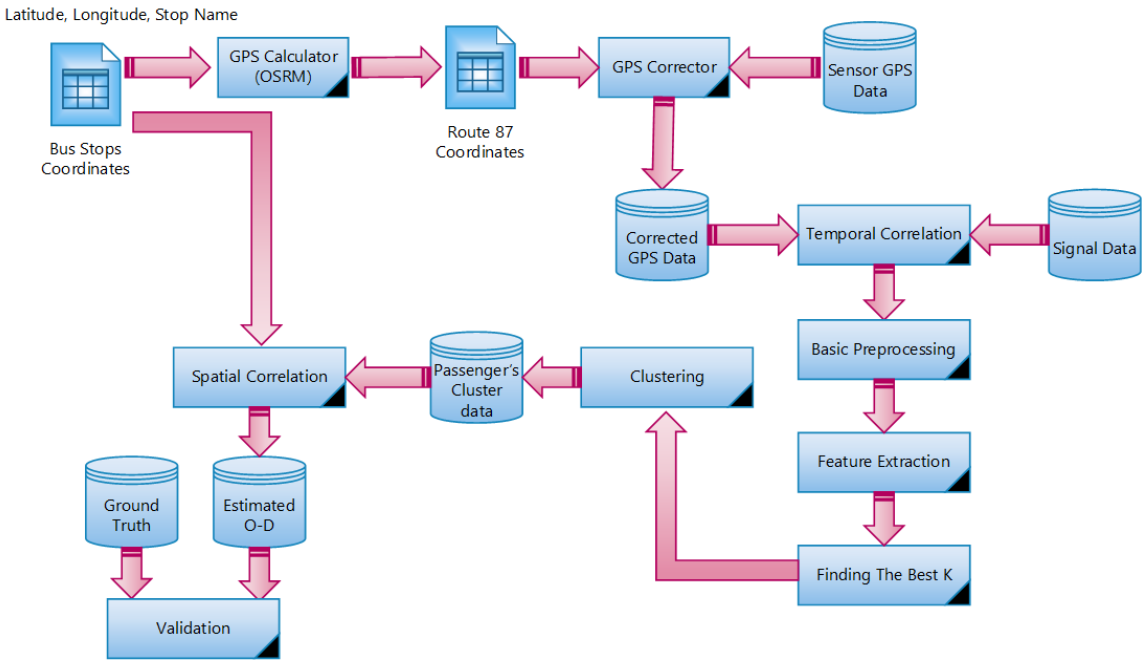

Figure 3.2 An overall view of the proposed clustering approach

The primary challenge in using GPS data is its accuracy. The captured GPS data points are approximate, and several factors such as weather condition and structural obstacles can affect the results. These factors will be discussed in details in section 3.2.4. Thus, before using the sensor's GPS data in this approach, we need to check the GPS data points and modify them where they are not accurate.

To do this, we first found the latitude and longitude of all bus stops which were on the bus route. Then, we used an open-source software to find the coordinates between these bus stops. In other words, we found all the GPS coordinates on the selected bus route. Finally, by comparing these coordinates and the sensor's GPS dataset, we corrected the GPS data errors.

Moreover, we matched signal records with corrected GPS data using temporal analysis. In other words, we assigned proper latitude and longitude to each signal record by comparing the timestamps of both datasets. If the exact timestamp match was missing in the dataset, we assigned the closest one.

The next step is a basic preprocessing. Since Wi-Fi and Bluetooth signals are inherently different and follow distinct standards, we separated Wi-Fi and Bluetooth

signals for further processing. Also, in this step, we removed clear outliers and noise from data sets. These outliers not only can affect the clustering results negatively but also do not provide valuable information for the purpose of this research. For these reasons, we defined basic filtering steps to eliminate noise and outliers. We then extracted features from the data sets to be used in the clustering algorithms.

K-Means and Hierarchical agglomerative clustering algorithms are considered for dividing data records into passengers and non-passengers. In our previous study on the performance of six different clustering algorithms, we found that K-Means and Hierarchical agglomerative clustering exhibited the best performance in this particular application (Afshari, Jalali, Ghods, & Raahemi, 2019). To obtain the optimal number of clusters, we used the Silhouette index which measures how well-separated the clusters are.

Having completed the clustering step to identify the "passengers" group, each passenger's first and last record coordinates were matched with bus stops coordinates to find their origin and destination. Then, by comparing the estimated O-D matrix with the ground truth data, we evaluate the performance of the proposed clustering approach. The following sections explain the methodology and its steps in details.

3.1. Data collection

3.1.1. Data collection device

We use SMATS Traffic Solutions' sensor (SMATS TrafficBox™), for collecting Bluetooth and Wi-Fi signals of bus passengers' mobile devices in order to estimate the Origin-Destination matrix. SMATS Traffic Solutions is our industry partner in this research.

Figure 3.3 SMATS TrafficBox™ Sensor (SMATS, n.d.)

SMATS TrafficBox™ sensor is a portable device which consists of Wi-Fi and Bluetooth scanners. It captures and stores signals from mobile devices along with the GPS module, battery, and the central processing unit. Linux operating system is used to run the sensor's software which initializes, captures, stores data on a local database. The sensor can run on battery for up to 24 hours. Figure 3.3 shows an actual SMATS TrafficBox™ sensor. Detailed information about this sensor can be found in Table 3.1.

Table 3.1 Technical Specification of SMATS TrafficBox™ (SMATS, n.d.)

Item	Value
Bluetooth Classic Module	-93dBm RX
Wi-Fi Module	+18dBm TX power (Max), -92dBm RX 802.11 b/g/n
Wi-Fi & Bluetooth Antenna	Omnidirectional - <300m (outdoor)
GPS Module	SiRF Start 4, -163dBm tracking sensitivity, 48 Track channels
Central Unit Processor	Quad-core ARM Cortex-A7 (armv7)
Central Unit Memory	1GB
Central Unit Storage	16GB (MicroSD)

This sensor passively listens to signals in an omnidirectional manner and collects Wi-Fi and Bluetooth MAC addresses, signal strength, the timestamp of the detection, and the GPS data of the sensor. MAC addresses (which are unique identification numbers for devices) are encrypted by the sensor using an irreversible hash function before storing the data in its memory. The Anonymization of the MAC addresses is a security and privacy

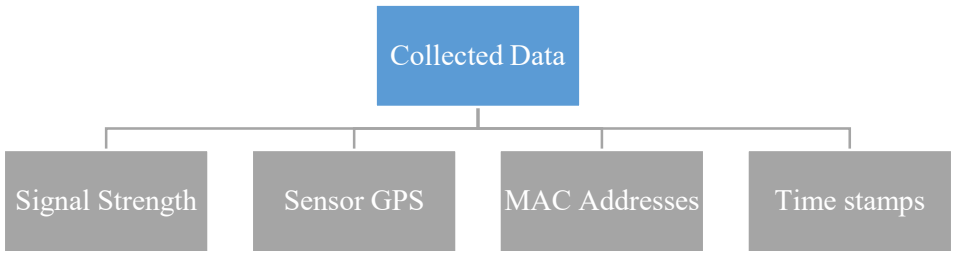

Figure 3.4 The sensor collected data

issue that was implemented inside the TrafficBox sensor to eliminate any privacy concern regarding the anonymity of the captured data from passengers' devices. Therefore, it is mathematically impossible to recover the original MAC address of the passengers' device from their respective hash version. This issue was clearly communicated with the Ethics Board of the University of Ottawa, and an Ethics approval is secured from the University of Ottawa Ethics Board for this research (Please see Appendix A for a copy of the Ethics Certificate). Also, the anonymized MAC addresses will not be reported anywhere. This

research project concentrates on passengers' flow patterns and not the MAC address itself. In addition, the collected data were stored and analyzed using the researcher's computers, and each of these computers has a username and password that prevents others from accessing the data. After completing the research, the data will be removed from computers and the SMATS TrafficBox™ sensor's memory card. Figure 3.4 shows the types of collected data using SMATS TrafficBox™ sensor.

3.1.2. Data Collection

To design the right setting for data collection, we first conducted several pilot tests with SMATS TrafficBox™ in Ottawa, ON. This helped us to avoid wasting our effort and time on experiments with improper settings. Also, these pilot studies helped us to learn how the sensor works in different situations, and what types of data it generates.

The first experiment was designed to investigate the difference between the RSSI (Received Signal Strength Indicator) of the signals inside and outside of the bus. For this purpose, we defined two scenarios: the first scenario was using only one mobile device on the bus, and the second one was using more than one mobile device on the bus. For both scenarios, Wi-Fi and Bluetooth of the mobile devices were turned on. Since we knew the MAC addresses of our mobile devices, we used the same hash algorithm that is being used in the SMATS TrafficBox™ device to determine the hash strings of our devices. By matching our devices hashed MAC addresses with the collected data, we found and analyzed their related records. Also, we compared the devices' RSSI with each other. By comparing the RSSI results, we observed that the average RSSI values of devices inside the bus are greater than -60 dBm. In addition, when the mobile devices' screens are on, they regularly send probe requests to find nearby access points, and consequently, the sensor will detect them more often. In other words, their number of detections are higher than the others.

The second experiment was designed to investigate the feasibility of conducting this research on different bus types (40-feet, 60-feet, and double-decker buses), with various locations selected for the sensor. For this purpose, we scheduled a trip for each of these bus types and placed the sensor at the same distance from the front and rear

passengers. Moreover, since we wanted to collect ground truth data while doing the main experiment, it was essential to consider the possibility of manually recording passengers' Origin-Destination.

- *40-feet bus:* This bus type has two doors (one at the front and one at the back). Only the front door is used for boarding passengers because the back door does not have a fare box. However, both doors can be used for alighting. This bus type is the best one for conducting the main experiment due to its size and number of doors which makes it possible to accurately record the O-D data of the passengers for further validation (i.e., ground truth data).
- *60-feet bus:* This bus type has three doors which commuters can use for both boarding and alighting. Its three doors and size make the observation and recording passengers' movement difficult and impractical. Therefore, we did not conduct the research on this type.
- *Double-decker bus:* This bus type like the 40-feet bus has two doors and in this model, both doors are used for boarding and alighting passengers. It also has two levels which makes it impossible to collect ground truth data since an observer cannot see the passengers of the other level. In addition, due to the distance and the ceiling, placing the sensor on the first level might affect received signal strengths from the second level. Thus, we did not conduct the research on this model either.

After conducting several experiments with different settings, we designed the main experiment of this research. Since commuters that use public transit are more frequent during the rush hours, we decided to perform the experiment during the evening rush hours from 4:00 PM to 5:30 PM to collect enough data. Among different urban routes in Ottawa, we selected route 87-Baseline because it serves downtown Ottawa which is a crowded area and mostly 40-feet buses operate on this route. Also, this route has 79 bus stops which it takes approximately one hour and a half to commute from the first stop to the last one. Figure 3.5 shows route 87 bus stops on the map.

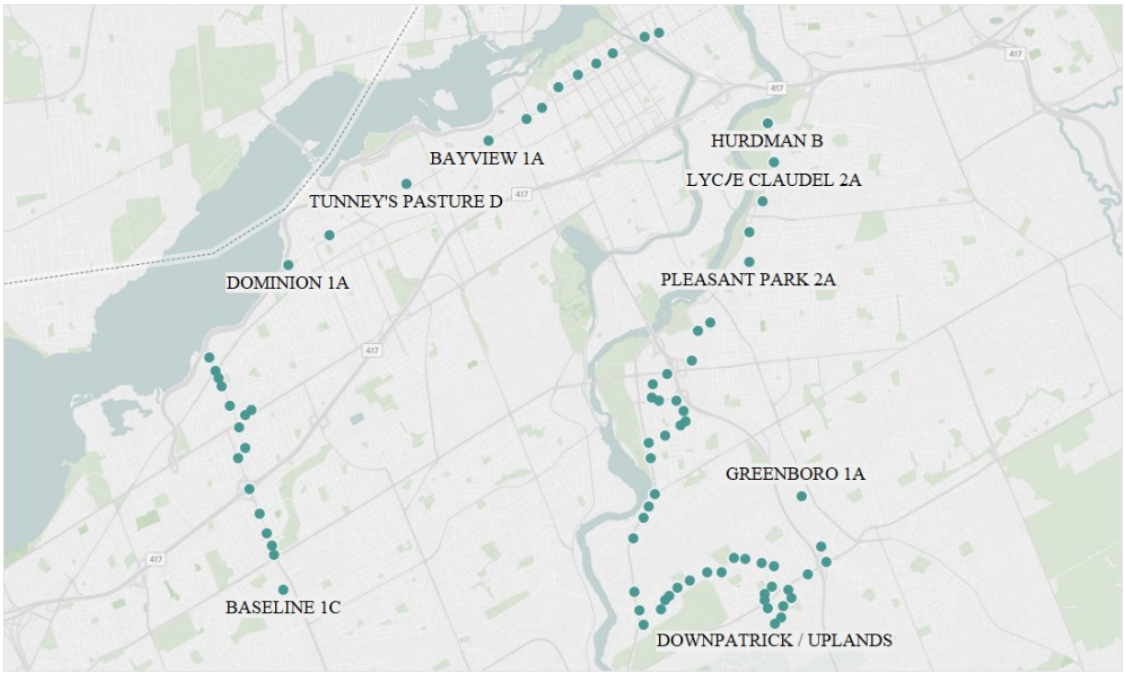

Figure 3.5 Bus stops on Route 87

We conducted the main experiment in six days during evening peak hours (4:00 to 5:30 PM). The SMATS TrafficBox^{TM} sensor was placed in the middle of the bus to capture Wi-Fi and Bluetooth signals of surrounding mobile devices with 1-second resolution. Moreover, the sensor was configured only to capture Wi-Fi probe frames and discard other Wi-Fi frames because the bus passengers do not connect to any access points and their mobile devices can only send probe requests. Collecting only probe frames help to filter noise which is received from the access points and other surrounding devices which are connected to the nearby access points. Figure 3.6 shows collected Wi-Fi and Bluetooth signals for the first trip.

All along the trips, two observers were onboard to collect origin and destination locations of bus passengers manually (i.e., ground truth data). We used the ground truth data in chapter 5 to validate the final results.

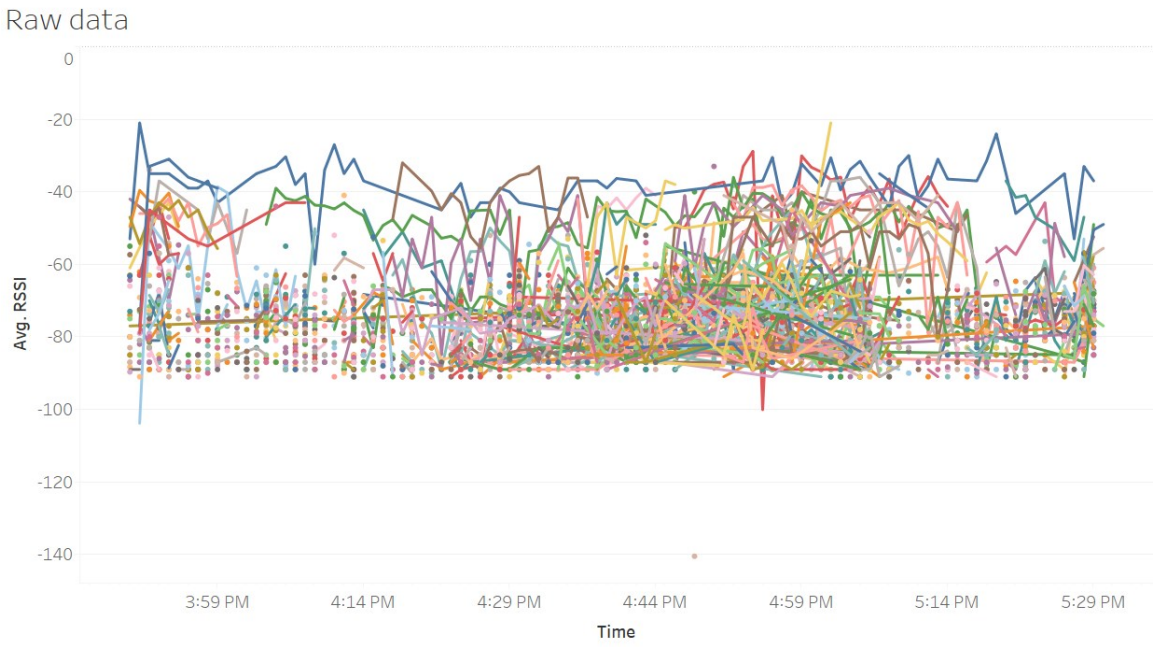

Figure 3.6 Wi-Fi and Bluetooth raw signals (different colors represent different MAC addresses)

3.2. Data Preprocessing

Data preprocessing is an essential process before analyzing the dataset since real datasets are usually noisy and incomplete. Preprocessing is a tedious task which takes significant time from researchers in the Machine Learning projects. This process consists of several steps such as data cleaning, data filtering, data transformation, data reduction and so forth (Kotsiantis, Kanellopoulos, & Pintelas, 2006). In this research, we used Python 3.6 programming language and Pandas and Numpy libraries for implementing our proposed methodologies.

3.2.1. Data Cleaning

Our collected dataset contains a significant amount of noise records because the TrafficBox™ sensor collects all the surrounding Wi-Fi and Bluetooth signals which are within the range of its antennas including the passengers' signals. In other words, the signals coming from the adjacent buildings, motorists, bicyclists, and pedestrians are also captured in the dataset, which makes it difficult to distinguish passengers' signals from

non-passengers. Therefore, we need to apply several preprocessing methods to remove obvious outliers and incomplete data.

The first preprocessing method used in this research is identifying and removing outliers. For this purpose, we define some soft thresholds to filter out data records which are significantly out of range. Since we want to use K-Means and Hierarchical clustering methods to find passengers' and non-passengers' clusters automatically, we only remove some apparent outliers and incomplete data at this stage.

For estimating the O-D data of passengers, there should be at least two records for each MAC address. The first record indicates the origin and the other one shows the destination. In certain cases, the sensor only captured one record for a MAC address which makes it impossible to find the O-D data or extract any valuable information from it. Thus, in this situation, we considered the record as an incomplete one and removed it from the dataset.

If the number of detection of MAC address < 2, then remove the record

The minimum time interval between two bus stops for the selected route is 50 seconds which means that the minimum trip duration is 50 seconds for this route. Since buses only stop at designated bus stations (commuters cannot get on and off the bus between the stops), we safely assume that if the trip duration for a specific MAC address is less than 50 seconds, this MAC address is related to a non-passenger mobile device. Thus, we remove it from the dataset.

If the Trip Duration < 50 seconds, then remove the record

As we mentioned before, the RSSI values of signals depend on the distance between transmitters and a receiver. The Wi-Fi and Bluetooth devices which are closer to the sensor have stronger signals than the other devices. Based on the initial experiments and individual observations in the data collection section (to set up the right configuration and environment for collecting the real data), we observed that records with average signal strengths of -60 dBm or weaker are probably outliers. However, we set this threshold to -70 dBm because we want clustering methods to remove outliers automatically without setting harsh thresholds. Since mostly there is more than one

record for each MAC address in the dataset, we calculate the average of signal strength for each MAC address, and then we remove all the instances of MAC addresses with signal strength averages below -70 dBm.

If the RSSI Average < -70 dBm, then remove the record

Besides, traffic congestion in some streets and especially in downtown is heavy during rush hours, and buses often have to stop in bus stations for several minutes. Also, when buses are ahead of their scheduled time, they have to wait longer at the bus stops to meet their schedule. In these cases, the sensor might collect additional noises from the surrounding environment with longer durations while buses are waiting in the bus stops. As a result, the "travel duration" filter cannot eliminate the noise due to the long duration associated with these cases of travel time. Thus, we defined another threshold by which we calculate the traveled distance of each passenger using the GPS location of the bus, then filter those MAC addresses that have zero travel distance.

If MAC addresses are detected multiple times AND the Travel Distance = 0, then remove the record

After applying all data cleaning steps, a significant amount of incomplete and outliers are filtered. Figure 3.7 shows the remaining Wi-Fi and Bluetooth signals for the first trip.

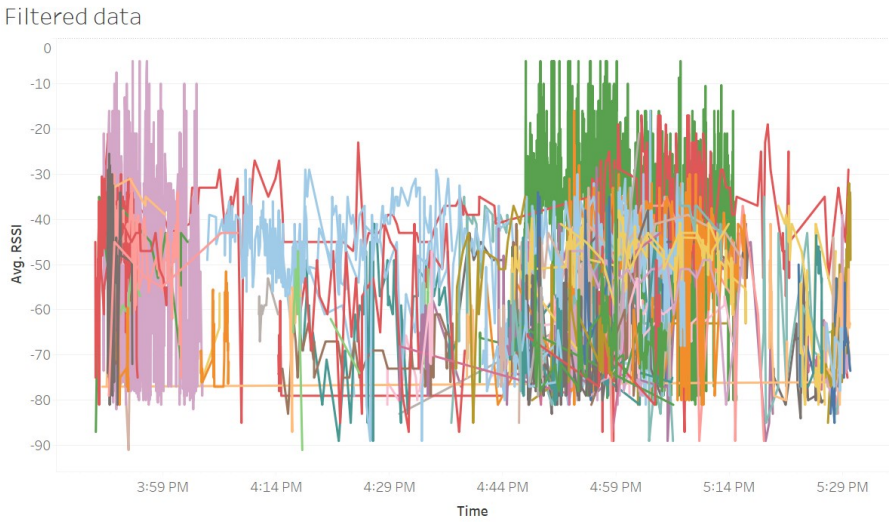

Figure 3.7 Filtered signals (different colors represent different MAC addresses)

3.2.2. GPS Modification

Global Positioning System (GPS), which provides accurate time and location information, consists of over 30 GPS satellites orbiting around the Earth. These satellites transmit radio signals including the satellite location and accurate time through space at the speed of light. GPS devices on the Earth receive these signals and also capture their precise received time. By multiplying signals' travel time to the speed of light, the distance between GPS satellites and GPS devices can be calculated. In order to obtain the location of a GPS device in three dimensions, the distance between at least four GPS satellites and that device should be computed. This method is called "Triangulation" (U.S. Government, 2016).

The accuracy of the GPS data relies on several factors such as weather conditions, signal obstructions which are caused by surrounding buildings, walls or trees, receiver quality, and exposure of the receiver to the sky (indoor or outdoor) (U.S. Government, 2017). Therefore, there is always some amount of error in specifying the location of a receiver, which we need to take this error into account as well.

Since we use the GPS data to estimate the locations of boarding and alighting of passengers in this research, knowledge of the accurate location of the bus at any given time is critical to assign correct locations to passengers. Each GPS record captured by SMATS TrafficBox™ has three attributes, namely, latitude, longitude, and timestamp. In order to assign a location to each passengers' record, we correlate the GPS data with the passengers' device records based on their timestamp. However, considering the inaccuracy of the GPS data in some specific areas such as downtown led us to implement an algorithm to efficiently modify the GPS data based on the bus route path. Figure 3.8 shows the captured GPS data points for a part of Route 87 in downtown Ottawa.

Figure 3.8 GPS data points with error (Route 87 downtown Ottawa)

In Figure 3.8, the red line indicates the bus path and the blue points shows the captured GPS data points. All the inaccurate GPS data points must be mapped to the bus path precisely.

In order to solve this problem, we employed the Open Source Routing Machine (OSRM) library written in C++. This library helped us to find the shortest path between a given origin and destination points using OpenStreetMap (OSM) data. By providing all bus stops coordinates which we obtained from the OC Transpo website, this library enables us to calculate all the GPS coordinates between the bus stops. Therefore, by knowing the coordinates of the bus stops, as well as their intermediary points, we could map all of the recorded GPS points to the closest coordinates on the bus path in order to correct the GPS errors. The below Pseudocode illustrates the algorithm that we developed for correcting GPS records.

```
logged_coordinates = # list of {gps_coordinate<lat,lng>, timestamp} logged by
the device (not accurate)
actual_coordinates = # a sorted list of GPS coordinations on the bus route path
                     # from the first station to the last station (retrieved from
OSRM project)


# correct the logged_cordinate according to the actual_coordinates
```

```
from_index = 0
FOR i, actual_coordinate in acutal_coordinates:
    next_actual_coordinate = actual_coordinate[i + 1]

    FOR j IN actual_coordinates.split(from_index):
       logged_coordinate = logged_coordinates[j]

        # if the next coordinate is closer => break and go to the next item
        IF distance(next_actual_coordinate, logged_coordinate) <
distance(actual_coordinate, logged_coordinate):
            from_index += j
            break

        ELSE:
            # set the correct coordinate
            logged_coordinates[j + from_index].lat = actual_coordinate.lat
            logged_coordinates[j + from_index].lng = actual_coordinate.lng

# now, logged_coordinates has corrected <lat,lng> and the corresponding
timestamp
```

Figure 3.9 shows the recorded GPS data versus the modified one. As it is shown, all the blue coordinates were corrected and mapped to the actual path of the bus.

Figure 3.9 Corrected GPS data for Route 87 downtown Ottawa

Moreover, it is possible that several recorded GPS data are mapped to one specific coordinate on the bus path. This issue happens when the bus is waiting at the red light, or the bus stop, or when the traffic is heavy. Therefore, by counting the number of coordinates that have been mapped to a specific point, we can infer the waiting patterns

of the bus during each trip. Figure 3.10 visualizes the wait time pattern of the bus (route 87 Ottawa) for the second trip, where larger circles indicate longer wait time.

Figure 3.10 The GPS coordinates count (larger circles indicate longer wait time)

3.2.3. GPS Assignment to Passengers' Records

GPS data plays an essential role in finding passengers O-D location. Each record in the GPS data consists of the corresponding timestamp indicating that when the data was recorded, its latitude, and longitude. Therefore, by matching passengers' records timestamp with GPS data, we can estimate the passengers approximate boarding and alighting locations. However, sometimes there is no exact timestamp match for passengers' records in the GPS data because the resolution of the GPS module and the Wi-fi and Bluetooth sensors are different. In this case, the corresponding location (latitude and longitude) of the closest timestamp from the GPS data will be assigned to the passengers' records.

To find the alighting and boarding locations of the MAC addresses in passengers' cluster(s), first, we needed to extract all the records related to a specific address. The first recorded data for each MAC address is assumed as the origin data, and the last record is considered as its destination. Table 3.2 illustrates the process of assigning latitude and longitude to each passengers' record.

Table 3.2 A sample GPS assignment

Wi-Fi Module Data				GPS Data		
Name	**Type**	**RSSI**	**Time**	**Latitude**	**Longitude**	**Time**
MAC Address 1	Wi-Fi	-47	16:41:44	45.412066	-75.6661755	16:41:44
MAC Address 1	Wi-Fi	-57	16:42:28	45.4116154	-75.6666304	16:42:28
MAC Address 1	Wi-Fi	-55	16:45:19	45.4152318	-75.6568165	16:45:20
MAC Address 1	Wi-Fi	-55	16:46:06	45.4156303	-75.656634	16:46:05
MAC Address 1	Wi-Fi	-51	16:46:51	45.4191868	-75.6550741	16:46:51
MAC Address 1	Wi-Fi	-63	16:47:33	45.4171388	-75.6620461	16:47:34
MAC Address 1	Wi-Fi	-57	16:49:07	45.4180796	-75.6764981	16:49:05

The first detection of MAC address 1, the upper yellow line shows the origin location and the lower yellow line indicates the destination location.

3.2.4. Feature Extraction

Since the sensor data is indexed with timestamps, we need to use time series analysis methods to extract useful and valuable information from the dataset. Time series clustering method consists of three different approaches known as raw data-based, feature-based, and model-based (Warren Liao, 2005). Based on the nature of the dataset, we selected the feature-based clustering approach because the other ones are not suitable when the time series data is noisy (Warren Liao, 2005).

In feature-based clustering approach, the raw data should be converted to a set of features. It means that the time-series data should be converted to feature vectors for the clustering methods to calculate the distances (or similarities) between these vectors to assign them to different clusters. For converting the dataset to a set of vectors to use feature-based clustering, we extracted a number of features from the input data. We considered the following five components for the feature vector:

1. **The average of RSSI**: Summation of all the RSSI values for a specific MAC address divided by its counts.
2. **The variance of RSSI**: Variance of all the RSSI values for a specific MAC address.
3. **The number of detection**: Number of all the records for a specific MAC address.

4. **Travel time**: Time difference between the last and first detection time.
5. **Travel Distance**: The traveled distance between boarding and alighting coordinates.

3.2.5. Data Normalization

For each feature vector, we considered five features which have different ranges and units. Distance-based clustering methods such as K-Means algorithm work based on the distances between data points. Different ranges and values can affect the measured distance and generate poor quality clusters. Data normalization is the solution to this problem increasing the accuracy of clustering methods.

In this research, we used a feature scaling method known as "unit vector" to normalize the features. In this method, the components of the feature vector were calculated in a way that the new feature vector has the length equal to one. Equation 1 shows the formula that we used for normalization.

If $\vec{x} = (x_1, x_2, \dots, x_n)$: (1)

$$|\vec{x}| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

$$\hat{x} = \frac{\vec{x}}{|\vec{x}|}$$

3.3. Clustering Methods

In this section, we explain how K-Means and Hierarchical Agglomerative clustering techniques are used to distinguish passengers' and non-passengers' signals. We selected these two algorithms because in (Afshari et al., 2019), we found that these two clustering methods have the best performance in identifying the passengers.

3.3.1. K-Means Method

K-Means is an unsupervised clustering algorithm which takes the number of clusters (K) as an input, then partitions the data into K different clusters. This algorithm aims to increase both the similarity among the members of the same cluster and the dissimilarity

among the members of the separated clusters (Dunham, 2006). In this method, the similarity is measured based on the Euclidean distance. Equation 2, illustrates the objective function that we use for K-Means clustering which is also known as the least squared error function.

$$Objective\ Function = \sum_{j=1}^{k} \sum_{i=1}^{n_k} \left\| x_i^{(j)} - c_j \right\|^2 \quad (2)$$

In this equation, k is the number of clusters, n_k is the number of data points in cluster k, c_j is the centroid of cluster j and $x_i^{(j)}$represents the data points in cluster j.

By choosing the number of clusters (K), the K-Means algorithm begins with randomly selecting K initial points as clusters centroids. Afterward, the distance between each data point and all centroids are calculated, and data points are assigned to the nearest centroid. Then, the algorithm updates the centroid of each cluster by computing the mean of its data points. These two steps are repeated for each iteration until the centroids remain the same (Warren Liao, 2005).

The main drawback of this method is the sensitivity towards outliers. In other words, since the K-Means algorithm measures the similarity based on the distance between members, outliers can affect the final results negatively (Dunham, 2006).

3.3.2. Hierarchical Method

Hierarchical Clustering has two methods which are Agglomerative (bottom-up) and Divisive (top-down). In Hierarchical agglomerative clustering, initially each data point is considered as one cluster, and this algorithm starts by calculating the distances between all clusters. Afterward, the two closest clusters are merged, and the distance matrix is updated. These steps are repeated until all data points are placed either in one cluster or the desired number of clusters (Warren Liao, 2005). In the divisive method, the order of the above steps is reversed, which means that all the data initially are considered as one big cluster, and then this method splits them into a specified number of clusters.

Single linkage, Complete linkage, Centroid distance, and Ward's minimum variance are four ways of measuring the distance between clusters to find the closest ones. In the single linkage method, the distance is calculated based on the closest member of each cluster, while in the complete linkage, the distance between furthest members is measured (Warren Liao, 2005) and then two clusters which have the smallest distance are merged. The centroid distance method measures the distance using the centroids of each cluster which also represents the members' average. In Ward's minimum variance method, the objective function is sum-of-squares variance. This method unifies the two clusters which produce the least sum-of-squares variance at each level (Warren Liao, 2005).

For illustrating the final clusters, the Hierarchical algorithm produces a Dendrogram diagram that represents the results as a tree. Figure 3.11 shows a sample dendrogram.

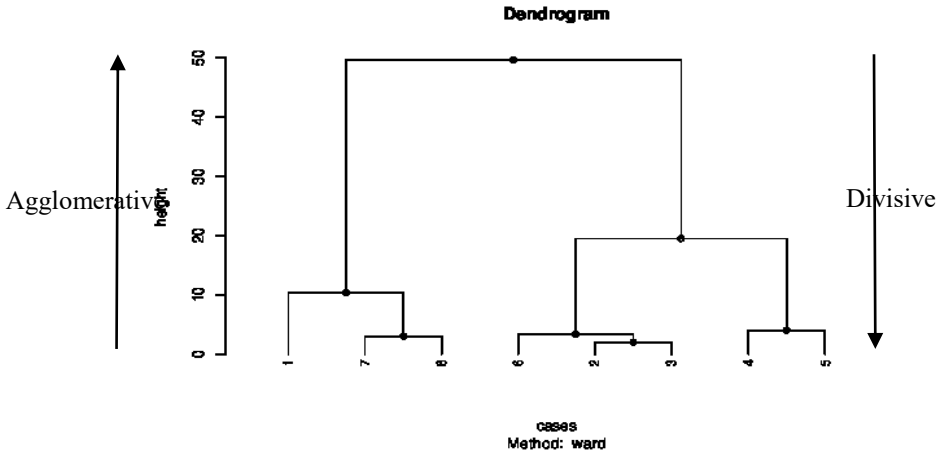

Figure 3.11 A sample dendrogram

3.3.3. Silhouette Index

Silhouette analysis measures the separation of final clusters. It can be calculated at both clusters and global levels. The result of this index is in the range of [-1, 1]. Values near to 1 indicate a better separation of the clusters while the results near to -1 show that data points might be allocated to the incorrect clusters. Equation 3 illustrates Silhouette index formula for the i-th data vector in the cluster $C_j,\ j = 1, \ldots, K$ (Petrovic, 2006).

$$s_i^j = \frac{b_i^j - a_i^j}{Max\{a_i^j, b_i^j\}} \quad (3)$$

Where b_i^j is the minimum average distance between the i-th data vector in the cluster C_j and all the vectors clustered in $C_k, k \neq j$, and a_i^j shows the average distance between the i-th vector in the cluster C_j and other vectors in that same cluster.

The Silhouette index at cluster level for cluster $C_j = \left\{X_1^j, X_2^j, \ldots, X_{m_j}^j\right\}$ can be obtained by equation 4.

$$S_j = \frac{1}{m_j}\sum_{i=1}^{m_j} s_i^j \quad (4)$$

Where m_j is the number of members in the cluter C_j.

Global Silhouette index is given by equation 5 (Petrovic, 2006).

$$S = \frac{1}{K}\sum_{j=1}^{K} S_j \quad (5)$$

Where K is the total number of clusters.

3.4. K-Means and Hierarchical Agglomerative Clustering Results

As mentioned earlier, both K-Means and Hierarchical Agglomerative clustering methods need the number of clusters (K) as an input. In order to find the optimal number of clusters for each of these methods, we employed Silhouette analysis for all the six trips, then examined the quality of clusters for different numbers of clusters from K=2 to K=5. Since we wanted to cluster and analyze the Wi-Fi and Bluetooth data separately due to their nature of signal differences, we calculated the Silhouette index for each type of signal exclusively. Table 3.3 and 3.4 shows the results for both Bluetooth and Wi-Fi data using Hierarchical clustering. The green cells indicate the optimal number of clusters for each trip.

Table 3.3 Silhouette index for Bluetooth data using Hierarchical agglomerative algorithm

Hierarchical Algorithm – Bluetooth				
Trips	**K=2**	**K=3**	**K=4**	**K=5**
First	0.720	0.641	0.630	0.604
Second	0.672	0.650	0.634	0.510
Third	0.798	0.814	0.765	0.758
Forth	0.710	0.729	0.733	0.753
Fifth	0.703	0.655	0.668	0.559
Sixth	0.665	0.700	0.715	0.717

Table 3.4 Silhouette index for Wi-Fi data using Hierarchical agglomerative algorithm

Hierarchical Algorithm – Wi-Fi				
Trips	**K=2**	**K=3**	**K=4**	**K=5**
First	0.632	0.539	0.512	0.503
Second	0.665	0.473	0.478	0.475
Third	0.603	0.501	0.510	0.510
Forth	0.537	0.566	0.591	0.518
Fifth	0.707	0.657	0.480	0.449
Sixth	0.687	0.608	0.643	0.579

Table 3.5 and 3.6 illustrate the Silhouette results for K-Means clustering.

Table 3.5 Silhouette index for Bluetooth data using K-Means algorithm

K-Means Algorithm - Bluetooth				
Trips	**K=2**	**K=3**	**K=4**	**K=5**
First	0.720	0.641	0.635	0.594
Second	0.672	0.650	0.619	0.604
Third	0.807	0.814	0.765	0.758
Forth	0.778	0.739	0.733	0.753
Fifth	0.703	0.603	0.638	0.560
Sixth	0.681	0.711	0.724	0.726

Table 3.6 Silhouette index for Wi-Fi data using K-Means algorithm

K-Means Algorithm – Wi-Fi				
Trips	**K=2**	**K=3**	**K=4**	**K=5**
First	0.630	0.547	0.524	0.556
Second	0.665	0.526	0.498	0.497
Third	0.603	0.574	0.563	0.567
Forth	0.576	0.594	0.591	0.532
Fifth	0.707	0.657	0.520	0.461
Sixth	0.698	0.683	0.643	0.579

Using the optimal number of clusters for each trip, we ran both K-Means and Hierarchical agglomerative clustering to distinguish passengers from non-passengers' signals. Once the clusters were created, we examined them and specified what each cluster represents. In other words, we labeled each cluster as either passengers or non-passengers based on the features' values of its members. Clusters' centroids could also be used to do this process. Figure 3.12 illustrates K-Means clustering results for the first trip Wi-Fi data.

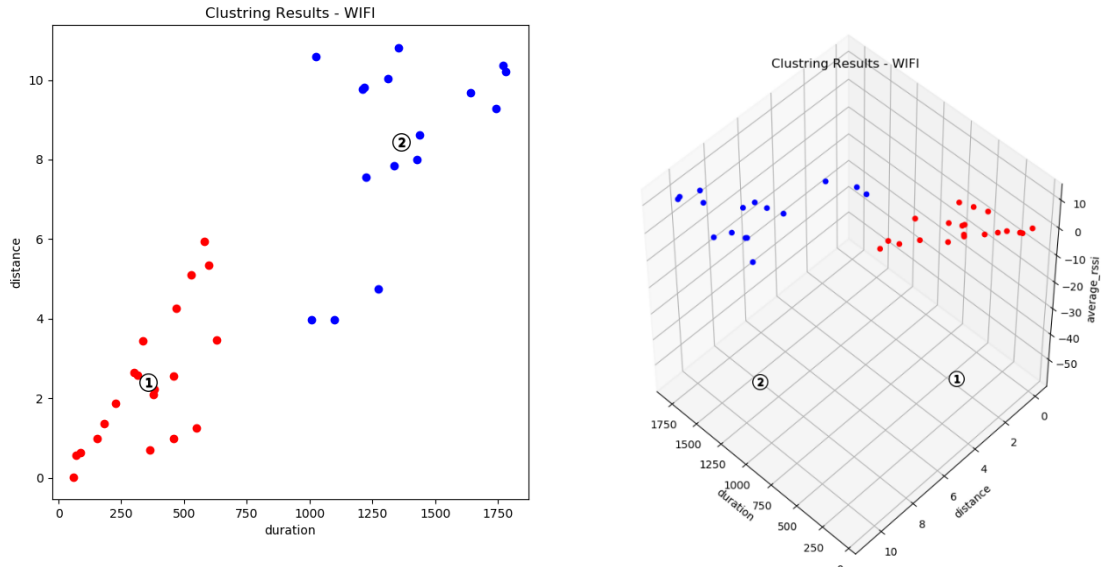

Figure 3.12 Clusters of the first trip Wi-Fi data

In Figure 3.12, two clusters of the first trip Wi-Fi data are shown. Onboard passengers have stronger signal strengths because they are closer to the sensor. Also, as they travel by bus, the sensor captures their signal for longer time and distances. Therefore, the MAC addresses with higher travel time, travel distance, and lesser RSSI average and variance values most likely belong to the actual bus passengers. As a result, we can infer what each cluster represents (passengers or non-passengers) by comparing the values of their centroids. For instance, when we have only two clusters, the cluster with higher travel time and travel distance, and lesser RSSI average and RSSI variance centroid values represent the passenger's cluster (the blue cluster in Figure 3.12), and vice versa. When the number of clusters is more than two, identifying the passenger's cluster(s) is harder because we cannot simply assign one cluster to the passengers and the other one to the outside people without further investigation. However, checking centroid points and comparing them with each other can give us a good insight into the clusters. Figure 3.13 shows the first trip Wi-Fi and Bluetooth signals after clustering.

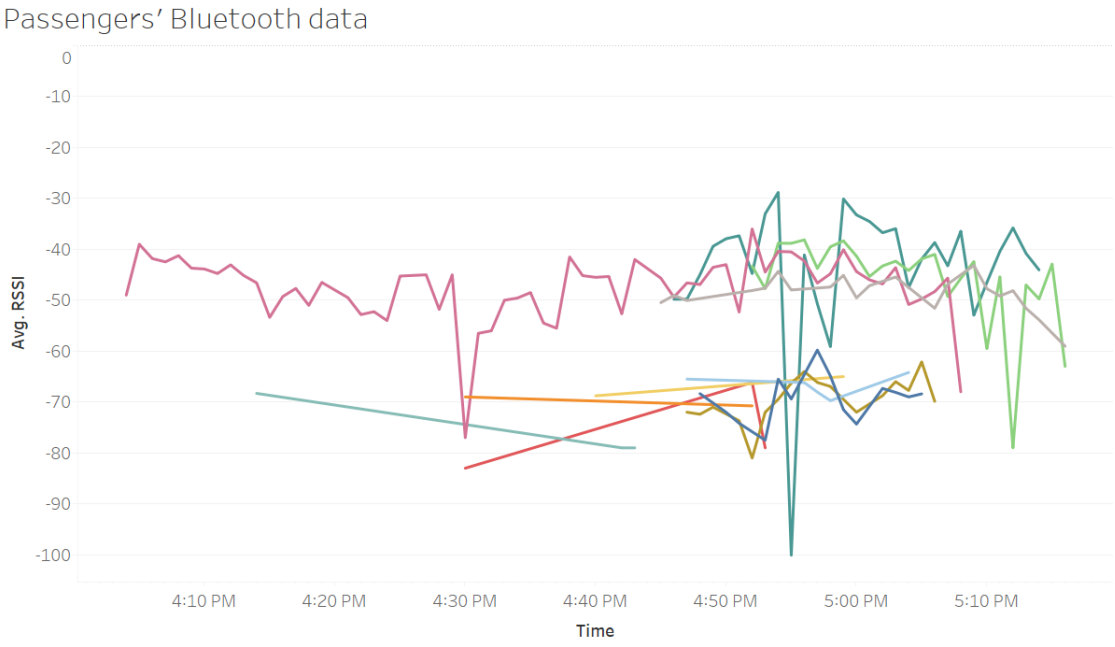

Figure 3.13 (a) Passengers' Bluetooth data of the first trip (different colors represent different MAC addresses)

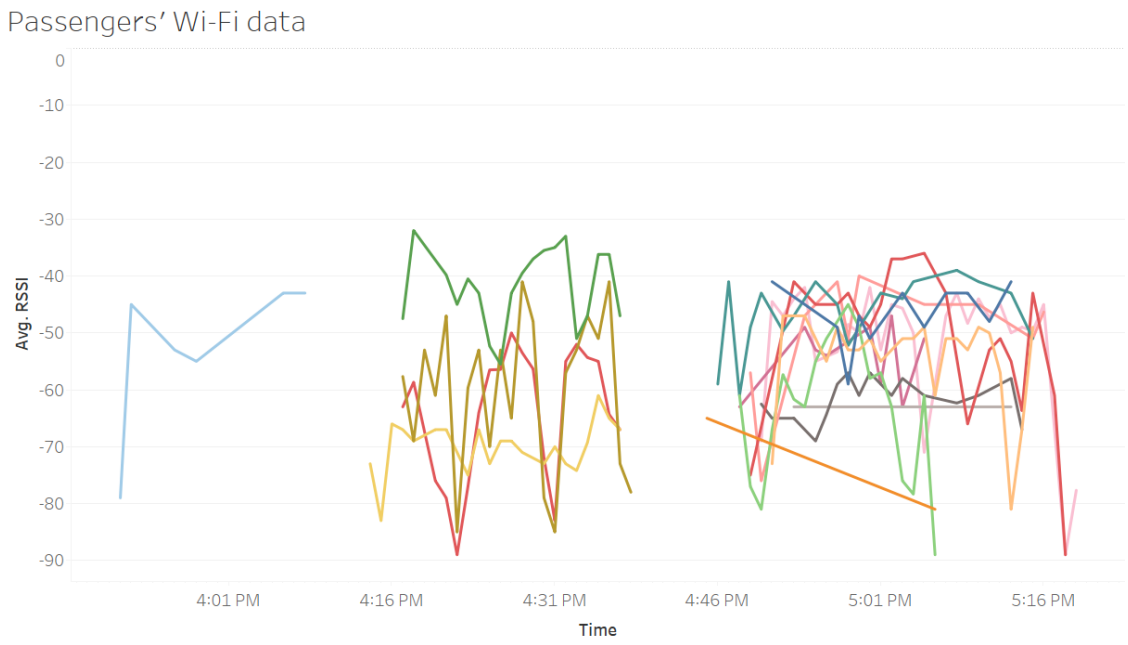

Figure 3.13 (b) Passengers' Wi-Fi data of the first trip (different colors represent different MAC addresses)

Figure 3.14 shows the comparison between the two clustering algorithms regarding the number of detected passengers.

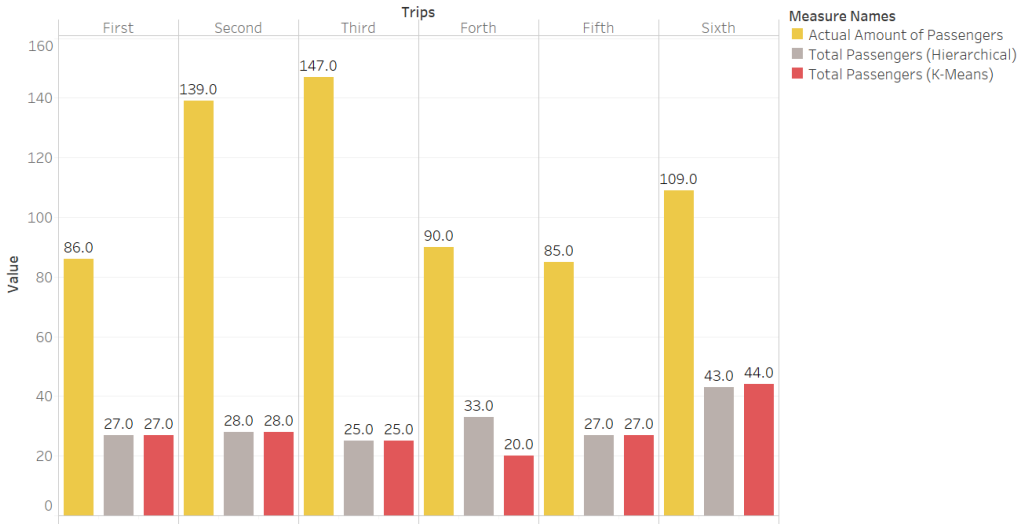

Figure 3.14 Actual versus detected passengers using two clustering methods

Table 3.7 shows the results of filtering, as well as both Hierarchical agglomerative and K-Means clustering. In this table, the actual numbers of passengers were obtained by two observers who were inside the bus during each trip. These observers counted the number

of passengers, and also, recorded their origin and destination locations. The filtered signal column indicates the number of Bluetooth and Wi-Fi MAC addresses which stayed in the dataset after the filtering process. Both Hierarchical agglomerative and K-Means columns show the passengers' cluster statistics. Finally, for each trip, the detection rate columns were calculated as shown in equation 6.

$$Detection\ Rate\ (\%) = \frac{detected\ passengers}{Actual\ passengers} \times 100 \qquad (6)$$

As it is shown in table 3.7, both Hierarchical agglomerative and K-Means clustering have produced almost the same results. Indeed, except for the fourth and sixth trips, these two methods detected the same number of passengers. Our analysis shows that the average detection rates for the Hierarchical agglomerative and K-Means methods are 29.39% and 27.14%, respectively.

Table 3.7 Clustering results

		Filtered Signals			Hierarchical Method			K-Means Method			Detection Rate (%)	
Trips	**Actual Number of Passengers**	**Wi-Fi**	**Bluetooth**	**Total**	**Wi-Fi**	**Bluetooth**	**Total**	**Wi-Fi**	**Bluetooth**	**Total**	**Hierarchical**	**K-Means**
First	86	26	65	91	17	10	27	17	10	27	31.39	31.39
Second	139	56	76	132	15	13	28	15	13	28	20.14	20.14
Third	147	52	58	110	15	10	25	15	10	25	17.00	17.00
Forth	90	24	72	96	16	17	33	14	6	20	36.66	22.22
Fifth	85	36	80	116	16	11	27	16	11	27	31.76	31.76
Sixth	109	44	72	116	14	29	43	16	28	44	39.44	40.36
										Average:	**29.39**	**27.14**

Chapter 4: Online Threshold-based Approach

Our second proposed approach is based on analysis of online sequential signals while applying specific thresholds. Unlike the first method, this approach analyzes the data in real time to differentiate between passengers and non-passengers' signals. This online approach facilitates public transit agencies' access to up-to-date information about passengers' origin and destination. Moreover, the transit agencies can employ this method for real-time analysis of the O-D matrix of the passengers as well.

In this approach, for each detected Wi-Fi or Bluetooth MAC address, the algorithm determines whether it is the first time the MAC address has been detected or not. If it is the first time, then proper boarding location will be assigned to it based on the GPS data (the GPS assignment process was explained in section 3.2.3). Otherwise, it will be buffered in the memory until the algorithm calculates the alighting time.

The frequency of transmitting Wi-Fi or Bluetooth signals is different for each handheld device and depends on various factors. For instance, if a passenger is working with his/her cell phone while the device Wi-Fi is on, the cell phone searches for nearby access points regularly, and sends probe requests periodically. However, if the mobile screen is off, the device rarely transmits probe signals. This issue might cause some errors in finding the destination of passengers since the only way to specify whether a passenger is still on-board or has been alighted, is to detect the signals transmitted by their devices.

To address this issue and analyze the data in real-time, the algorithm employs the following logic to estimate the alighting time of a passenger. When the bus arrives at a bus stop, the algorithm marks those passengers who are onboard and also has not been detected since the previous stop as a possible candidate for alighting. After a specific number of continuous failure to detect new signal from a passenger, the algorithm assumes that the passenger is no longer in the bus and assign the alighting location to its last detected record.

In this proposed algorithm, we considered three consecutive bus stops as the threshold on the number of continuous failure in detecting a passenger. In other words, if

the sensor has not detected a passenger for three consecutive stations, the passenger will be determined as alighted. Therefore, the results will be reported with three-station delay in this approach. The reason we chose the three-station threshold is that it is long enough to assure capturing signals even form idle mobile devices. As mentioned before, idle devices transmit Wi-Fi and Bluetooth signals less frequently than in-use devices. However, in some rare cases, the passenger might be still onboard and turns off his/her Wi-Fi and Bluetooth modules or switches his/her phone off. As a result, this situation is an unavoidable error.

Finally, since there is no clustering method to differentiate between passengers and non-passengers' signals in this method, we used the same thresholds as in section 3.2.1 to validate whether the detected MAC address is inside or outside of the bus. However, the RSSI average threshold is stricter than section 3.2.1 to eliminate additional noise.

4.1. General Threshold Setting

Similar to the clustering approach, the thresholds which were used for the validation process in this method are the number of detections, travel time, RSSI average, and travel distance. We set these thresholds as follow:

$$Number\ of\ detections > 2$$

$$Travel\ time > 50\ s$$

$$RSSI\ average > -60\ dBm$$

$$Travel\ distance > 0\ m$$

The number of detection is set to two because we need at least two records from a MAC address to determine its origin and destination. In other words, the first detected record determines its boarding location and the last one indicates its alighting place. Also, the time difference between passengers first and last detected record should be at least 50 seconds as the minimum time interval between two bus stops for the selected route is 50

seconds. Regarding the RSSI average, we set the threshold based on empirical (trial and error) method. To find the optimum point for this threshold, we tried different values (-60, -65 and, -70 dBm) for this feature. Based on the results, we realized that -60 dBm is the optimum value for this threshold as it can eliminate noise without removing that much onboard signal. Finally, travel distance threshold helps in deleting MAC addresses which are detected when the bus has to stop at bus stops or red lights.

In this method, other than Travel Time threshold which depends on the distances between bus stops of a specific route, other thresholds are generic enough also to be used for other urban routes. That is, we set fixed and general minimum required thresholds for identifying a passenger from the noise. The thresholds are general and not specific to route 87.

If a MAC address passes all of the thresholds, it is recognized as a passenger's MAC address; Otherwise, it is considered as noise and gets discarded. After this step, the algorithm determines the detected passengers O-D data which will be explained in details in Chapter 5 section 5.1. Figure 4.1 illustrates the flowchart of detecting passengers by analyzing their Wi-Fi and Bluetooth signals in real-time.

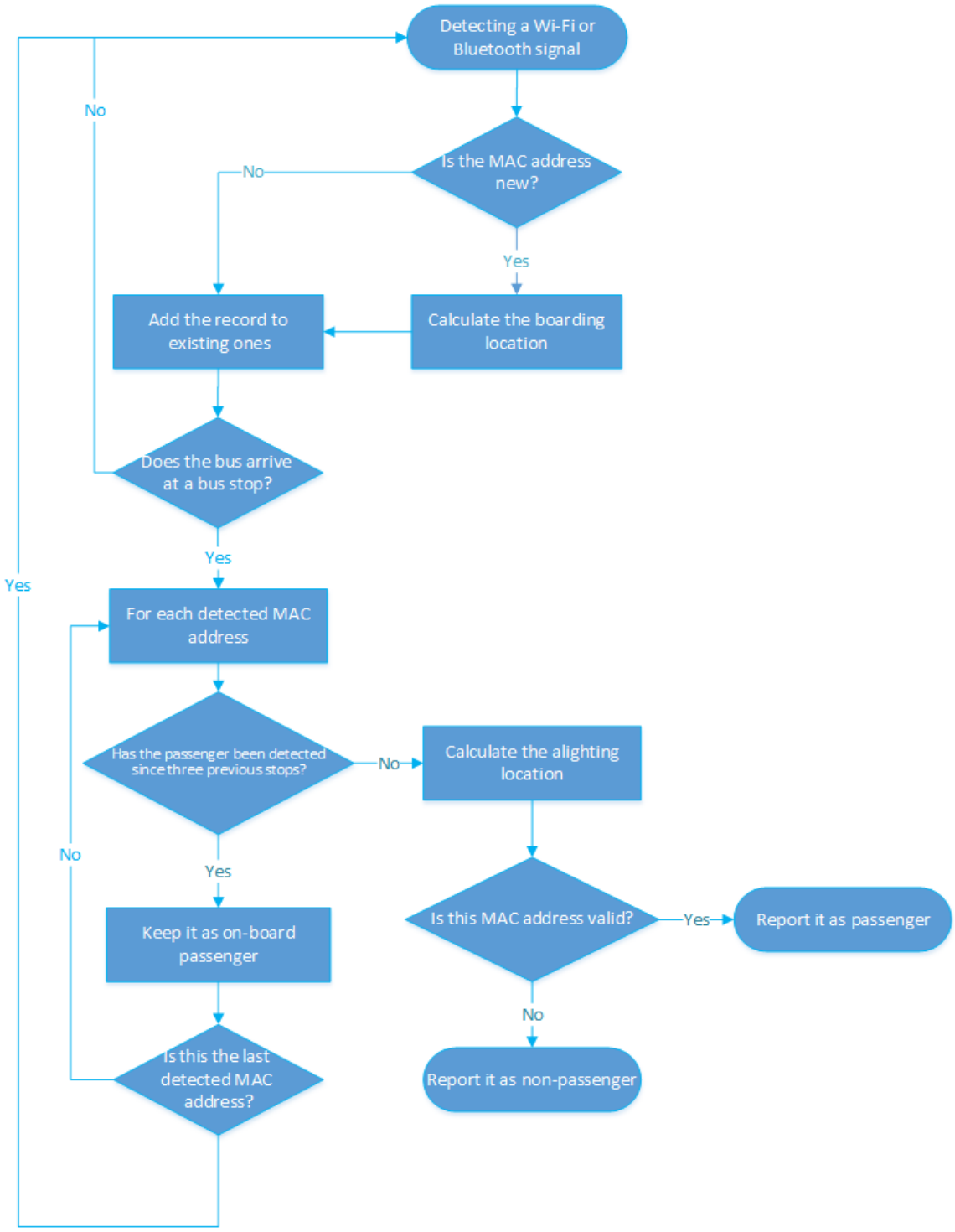

Figure 4.1 Flowchart of detecting passengers by analyzing their Wi-Fi and Bluetooth signals in real-time

4.2. Experimental Results

Table 4.1 lists the results of the online threshold-based analysis. In this table, the actual number of passengers which were obtained by two observers during each trip show how many passengers were onboard. The "Detected Passenger" column contains information about Wi-Fi and Bluetooth signals which were specified by the algorithm as a passenger, and finally, the "Detection Rate" values were obtained using equation 6. Also, the average detection rate for all trips is 22.78%.

Table 4.1 The results of the online threshold-based approach

		Detected Passenger			
Trips	**Actual Number of Passengers**	**Wi-Fi**	**Bluetooth**	**Total**	**Detection Rate (%)**
First	86	13	12	25	29.06
Second	139	15	12	27	19.42
Third	147	13	8	21	14.28
Forth	90	10	6	16	17.77
Fifth	85	21	11	32	37.64
Sixth	109	18	9	27	24.77
				Average:	**22.78**

Various factors can cause the difference between the actual and detected number of passengers. The first factor is that some commuters especially seniors do not either carry smart mobile devices or use their devices' Wi-Fi and Bluetooth modules when they are onboard. Moreover, passengers can freely turn on or off their handheld devices Wi-Fi and Bluetooth modules; Thus, these devices' signals might be considered as noise in the validation step. However, we could detect almost 23% of passengers during each trip which is a significant detection rate compared to other techniques. Based on the latest National Capital Region O-D survey performed by TRANS Committee in 2011 (Transportation-Committee, 2011), only 5% of National Capital Region population were interviewed for the survey which was considered as a rich sample.

Public transit agencies can estimate the actual number of passengers using the average detection rate with reasonable precision for each trip. For instance, the algorithm detected 25 passengers for the first trip. If we divide this number by the average detection rate of 22.78%, the total number of onboard people is estimated at:

Estimated number of passengers for trip #1 = $\frac{25}{22.78\%} = 109$

Actual number of passengers for trip #1 = 86

$$Estimation\ Error = \frac{|Actual\ number - Estimated\ number|}{Actual\ number} = 27\%$$

$$Accuracy\ of\ Detection = 1 - Estimation\ Error = 73\%$$

It is worth mentioning that the average detection rate in this calculation is only based on six trips. We need more data on many more trips (e.g., one month of trips) in order to have a better estimation of the average detection rate. The detection rate provides us insights to the passengers' usage of their Wi-Fi and Bluetooth modules. It is not, in any way, an accurate indication of the actual number of passengers on board. Nevertheless, our goal is to accurately estimate the O-D of the detected passengers, not the actual number of passengers onboard. This goal is achieved with a good accuracy as we will explain in Chapter 5.

Chapter 5: Estimation of Origin-Destination and Evaluation of the Results

5.1. Determining Origin-Destination Stops of Passengers

During each trip, the TrafficBox™ was configured to record GPS data with 2-second resolution. In other words, the device was recording the bus location every two seconds. As a result, the TrafficBox™ captured the bus location data all along the path. By assigning the GPS data to passengers' records, the locations of the first and last detection of passengers' signals were specified. These assigned locations are not necessarily near route 87 bus stops to estimate their O-D stops directly. Therefore, we had to employ a method to correlate these locations to their nearby bus stops based on Mishalani et al. (2016).

TrafficBox™ Wi-Fi and Bluetooth sensors can detect the signals which are within their antennas' rang. Thus, even before arrival of the bus at a bus stop, the device can capture Wi-Fi and Bluetooth signals from people who are waiting at the bus stop. In this case, although some passengers might get on the bus at that bus stop, the TrafficBox™ have already detected them in advance. Therefore, in the matching process, the boarding location which is assigned to their first record is not the location of the bus stop. Figure 5.1 illustrates this scenario.

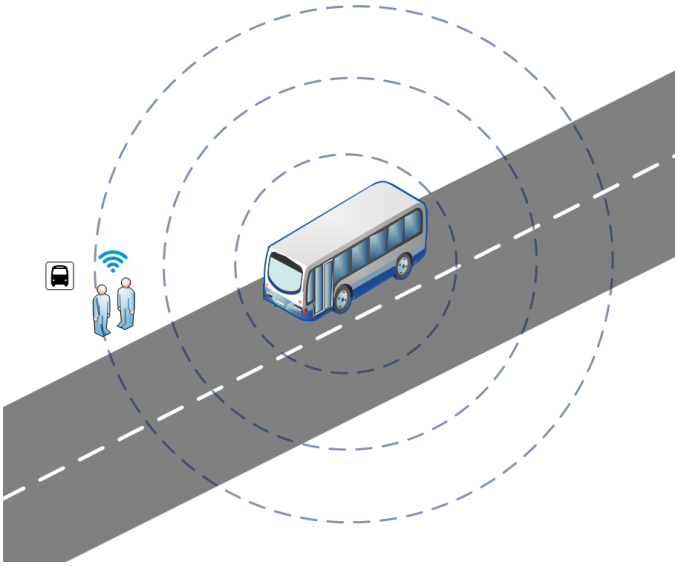

Figure 5.1 TrafficBox™ signal detection before the bus arrival at the bus stop

Similarly, even after the passengers get off the bus, the device can still detect their signals. In other words, when the bus leaves the station, TrafficBox™ can still capture the alighting passengers' signals as long as they are within the ranges of the sensors' antennas. Thus, this issue might affect the last detected signal of passengers and its corresponding alighting location. Figure 5.2 illustrates this scenario.

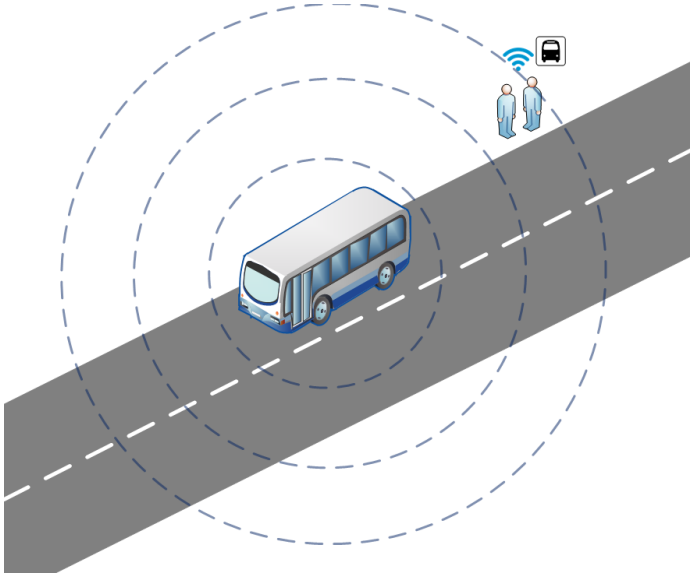

Figure 5.2 TrafficBox™ signal detection after the bus departure from the bus stop

To solve this problem, we set a distance threshold before and after each bus stop. This threshold was chosen based on the sensor antennas' range which is 50 meters. So if TrafficBox captures a signal in the radius of 50 meters around a bus stop, we specify that bus stop as either its origin or destination.

However, sometimes the first or last detected signals are neither close to bus stops nor within their radius of 50 meters. In this case, the first or last detected signals might be detected wherever between any two bus stops. For determining the O-D location of first detected signals, we considered the previous stop as the boarding location, and for the last detected signals, we assumed that the next bus stop would be the alighting location. Figure 5.3 illustrates this method using six sample signals which are shown with blue circles.

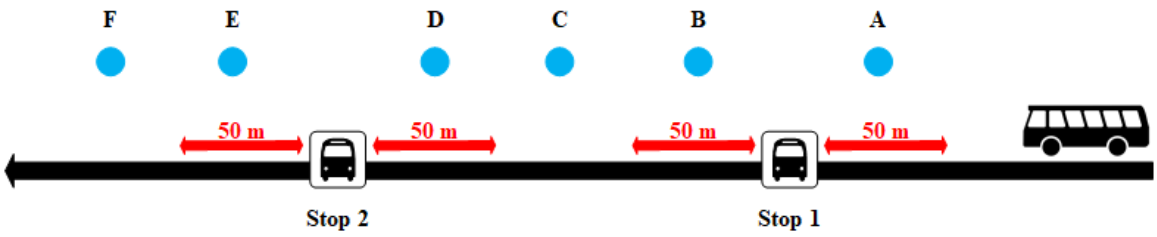

Figure 5.3 Different possible scenarios in determining O-D

If either signal A or B is the first detected signal of a passenger, it can be assumed that the passenger has boarded at Stop 1, because these signals are within the distance threshold of the bus stop. The reason that we consider signal A is boarded at Stop 1 is that TrafficBox™ can detect signals of waiting passengers at the bus stop before the bus arrives. However, this reasoning does not apply for signal B, but we safely assume that the passenger has turned on his/her mobile device's Wi-Fi or Bluetooth reception after getting on the bus.

If signal C is the first detected signal of the passenger, since it is not within 50 meters of neither Stop 1 nor Stop 2, we cannot assign it to them. In this case, we consider that the passenger was boarded at the previous stop which is Stop 1 and did not use his device's Wi-Fi or Bluetooth until point C. Since we do not have any more information about this passenger, this assumption is the best possible one.

If either signal D or E is the last detected signal of the passenger, it can be considered that the passenger has alighted at Stop 2, because these signals are within the distance threshold of the bus stop. For signal D, we can assume that the person stopped using his/her mobile device before arriving at the bus stop and as a result, the device did not capture his cell phone signals any longer. The assumption for signal E has the same explanation as signal A. Since TrafficBox™ can detect signals from far away, even after the bus leaves the bus stop, the device captures alighted passengers' signal until they are within its range.

If signal F is the last detected signal of the passenger, since it is not within 50 meters of Stop 2, we cannot assign the signal to it. In this case, we can consider that the passenger has alighted at the next stop which is Stop 3, and stopped using his device Wi-

Fi or Bluetooth after point F. Like signal C, because we do not have any other records from this passenger, it is the only possible assumption.

The following pseudocode shows how this procedure works for the first and last detected signals.

```
# the device captured from a unique fingerprint(mac address)
# type: 'IN' or 'OUT'
# data_point: <coordinate, timestamp>
# route_coordinates: list of all coordinates within the bus route

FUNCTION find_closest_station(data_point, type):
    target_coordinate_index = 0

    # try to find the passenger coordinate inside the route path coordinates
    FOR i IN route_coordinates:
        IF route_coordinates[i] EQUALS TO data_point.coordinate:
            # found!
            target_coordinate_index = i
            break

    next_station_distance = 0
    previous_station_distance = 0

    coordinate = route_coordinates[j]
    # find the next station
    FOR j IN RANGE(target_coordinate_index, LENGTH(route_coordinates)):
        # reached the next station!
        IF is_station(route_coordinates[j]):
            next_station = route_coordinates[j]
            BREAK
        # calculate distance between two points
        next_station_distance += distance(coordinate, route_coordinates[j + 1])

    j = target_coordinate_index
    coordinate = route_coordinates[j]
    # find the previous station
    WHILE j > -1:
        # reached the previous station
        IF is_station(route_coordinates[j]):
            previous_station = route_coordinates[j]
            BREAK
        # calculate distance between two points
        previous_station_distance += distance(coordinate, route_coordinates[j -
1])
        coordinate = route_coordinates[j]
        j = j - 1

    # now based on the type of boarding or alighting, we can decide which one
should be selected
    IF type EQUALS TO 'IN':
        IF next_station_distance < previous_station_distance AND
next_station_distance < DISTANCE_THRESHOLD:
            RETURN first_station
        ELSE:
            RETURN previous_station
    ELSE IF type EQUALS TO 'OUT':
        IF next_station_distance > previous_station_distance AND
previous_station_distance < DISTANCE_THRESHOLD:
            RETURN previous_station
```

```
        ELSE:
            RETURN next_station

# function to calculate the direct ground distance between two GPS coordinates
FUNCTION distance(point1, point2):
    AVG_EARTH_RADIUS = 6371  # in km
    MILES_PER_KILOMETER = 0.621371
    NAUTICAL_MILES_PER_KILOMETER = 0.539957

    # unpack latitude/longitude
    lat1, lng1 = point1
    lat2, lng2 = point2

    # convert all latitudes/longitudes from decimal degrees to radians
    lat1, lng1, lat2, lng2 = map(radians, (lat1, lng1, lat2, lng2))

    # calculate distance
    lat = lat2 - lat1
    lng = lng2 - lng1
    d = sin(lat * 0.5) ** 2 + cos(lat1) * cos(lat2) * sin(lng * 0.5) ** 2
    distance = 2 * AVG_EARTH_RADIUS * asin(sqrt(d))

    RETURN distance # in kilometers
```

5.2. Origin-Destination Matrix Creation

After assigning GPS locations to passengers' records, producing the Origin-Destination (O-D) matrix is the next step. O-D matrix is a square matrix $(n \times n)$ where n represents the number of bus stops in a route. Also, each row and column in this matrix indicates origin and destination stops respectively. By knowing each passenger's boarding and alighting stops, we could fill out the O-D matrix by adding one to the value of the cell $C_{Origin,Destination}$.

O-D matrix is commonly a sparse matrix. Each trip stop-to-stop O-D matrix contains a noticeable amount of zero values especially when there is plenty number of stops; because in this case, the size of the O-D matrix size is much higher than the total number of passengers. Also, the size of this matrix can complicate the accurate estimation of stop-to-stop passengers' movement (McCord, Mishalani, & Hu, 2012). That is why some studies such as Mishalani et al. (2016) aggregate all the passengers' data for all trips to obtain route-level O-D matrix. Indeed, route-level matrix sums up valuable ridership patterns which can be used for efficient bus planning and route design (McCord et al., 2012).

Moreover, grouping near bus stops is another way of reducing the size of the O-D matrix and improving the O-D estimation which was proposed by McCord et al. (2012). These authors purposed and validated two efficient heuristic algorithms for grouping bus stops. However, other transportation agencies such as the Transportation committee of Ottawa–Gatineau (TRANS) applied the different method for grouping stations. They merely divided different parts of the national capital region to distinct zones and considered all the stops in each zone as one (Transportation-Committee, 2011). In this way, not only has the size of the O-D matrix been reduced but also passengers' flows can be estimated more accurately.

We also grouped our selected route stops based on TRANS committee zones. Generally, TRANS divided the national capital region to 26 different zones which seven of them are rural areas. Figure 5.4 shows these zones on the Ottawa-Gatineau map.

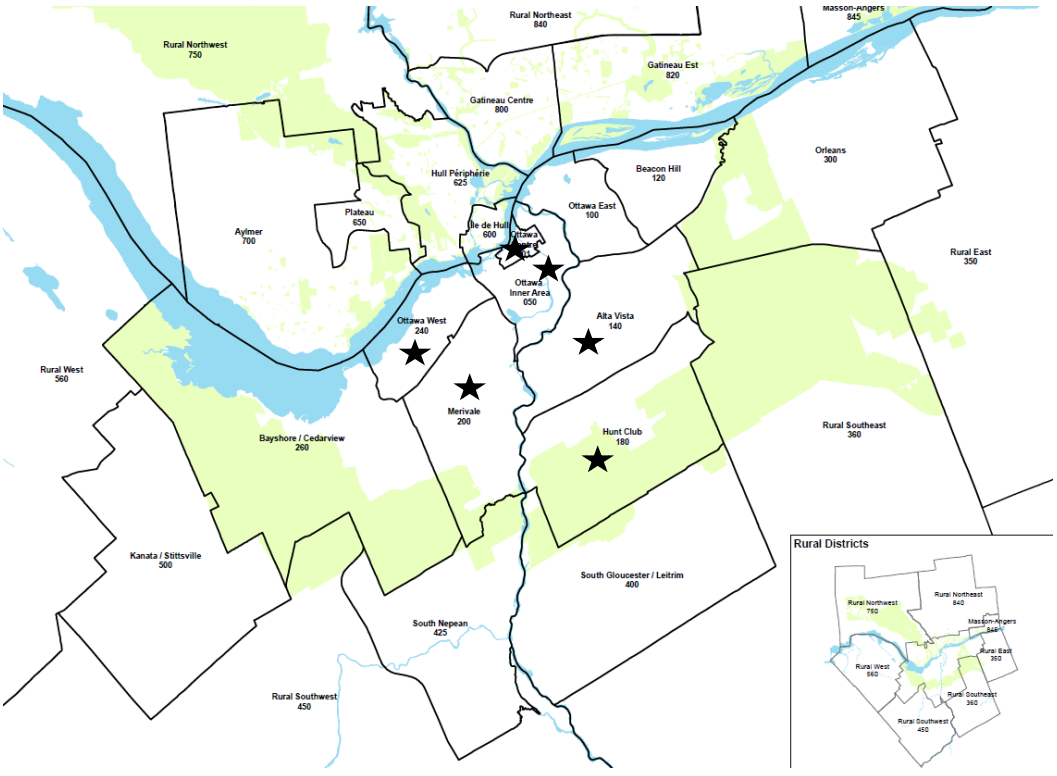

Figure 5.4 Different TRANS zones (Transportation-Committee, 2011)

In Figure 5.4, all the zones which route 87 serves have been marked with a black star. These six zones are Hunt club, Alta Vista, Ottawa Center, Inner Ottawa, Ottawa West, and Merivale. All 26 transit zone names and numbers can be found in Appendix C.

Grouping bus stops of route 87 into six zones reduced the O-D matrix size significantly which made estimating passengers' O-D and flow patterns easy. It also increased the estimated O-D matrix accuracy because finding precise stop-to-stop passengers' flows is not feasible. Table 5.1 and 5.2 show the actual and estimated O-D matrix, respectively, for the sixth trip.

Table 5.1 The actual O-D Matrix for the sixth trip

Origin \ Destination	Hunt Club	Alta Vista	Ottawa Center	Inner Ottawa	Ottawa West	Merivale
Hunt Club	42	2	0	0	0	0
Alta Vista	0	5	10	1	1	2
Ottawa Center	0	0	2	1	33	1
Inner Ottawa	0	0	0	0	5	0
Ottawa West	0	0	0	0	2	2
Merivale	0	0	0	0	0	0

Table 5.2 The estimated O-D Matrix using Hierarchical agglomerative Method for the sixth trip

Origin \ Destination	Hunt Club	Alta Vista	Ottawa Center	Inner Ottawa	Ottawa West	Merivale
Hunt Club	2	0	0	0	0	0
Alta Vista	0	0	9	1	1	1
Ottawa Center	0	0	0	0	21	1
Inner Ottawa	0	0	0	0	6	0
Ottawa West	0	0	0	0	0	1
Merivale	0	0	0	0	0	0

Since for all the trips, we only performed the experiment from GreenBoro station in Hunt Club zone to Baseline station in Merivale zone (one-way trip), the matrices in Table 5.1 and 5.2 are upper triangular. In round-trip experiments, all cells in the O-D matrix assume values. Also, values in the main diagonal which have the same origin and destination represent the number of passengers' flows within a zone.

5.3. Passenger Flow Patterns

Using each trip O-D matrix, we can analyze passengers' demands for traveling between different zones. In other words, each cell in the O-D matrix demonstrates passengers' flows between relative transit zones. Since all trips are performed during the same period, usually aggregating all the O-D matrices can give better insight into passengers' movements. Table 5.3 illustrates the aggregated O-D matrix based on threshold-based method results.

Table 5.3 Aggregated O-D matrix based on the threshold-based method results

Origin \ Destination	Hunt Club	Alta Vista	Ottawa Center	Inner Ottawa	Ottawa West	Merivale
Hunt Club	37	4	0	0	0	0
Alta Vista	0	10	20	2	0	0
Ottawa Center	0	0	11	9	32	2
Inner Ottawa	0	0	0	1	9	0
Ottawa West	0	0	0	0	10	3
Merivale	0	0	0	0	0	1

Also for better understanding the passengers' ridership patterns, we can plot their movement on the map (Figure 5.5). The thickness of the arrows indicates the number of flows. This plot provides public transit agencies with an overall visualized information about riders' demands.

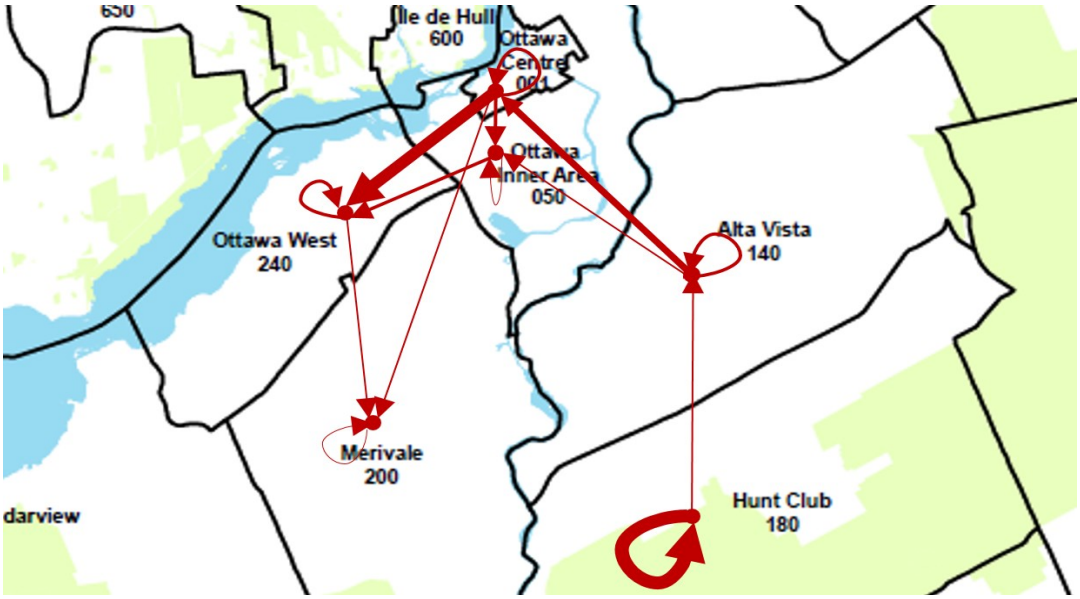

Figure 5.5 Passengers' movements for all trips (the thickness of the arrows indicates the number of flows)

As it is shown in Figure 5.5, the highest number of passengers' flow is within the Hunt Club zone. Route 87 starts from GreenBoro station in Hunt Club where the last stop of Ottawa Train is located. Thus, passengers who get out of the train in that station mostly use route 87 for short trips within the Hunt Club zone, and this issue justifies this pattern.

Also, results indicate that passengers tend to commute from Ottawa Center to Ottawa West in evening peak hours. There is a meaningful reason for this flow. Since many people are working in Ottawa Center zone (Downtown Ottawa), they start to leave their work around 4:00 until 5:00 PM. As a result, they use route 87 for going home. Probably passengers' flow in the morning rush hours is reverse, and it is from Ottawa West to Ottawa Center. Also, the results show that some people commuted from Alta Vista zone to Ottawa Center where some important places such as Ottawa Parliament Hill, Rideau Center Shopping Mall, University of Ottawa, and main Ottawa Public Library are located. Therefore, it can be inferred that the reasons for this flow might be leisure and shopping.

5.4. Evaluation of the Proposed Approaches

For evaluating experimental results and calculating the error, there are several formulas such as Mean Absolute Error (MAE), Root Mean Square Error (RSME), and Mean Absolute Percentage Error (MAPE). In these methods, a predicted or experimented value is compared with an observed one. However, in this study, the sensor could detect a subset of passengers not all of them for each trip so we actually could not compare the estimated O-D matrix with actual ones to find the errors. Indeed, such a comparison only produces a high error since the sensor cannot capture passengers who do not have a mobile device or do not use it during the trip. As a result, the passengers who are on-board but not detected are considered as errors mistakenly. Indeed, in this application, the sensor can only detect a sample of passengers not all of them.

In this study, the best way to obtain error can be calculating the percentage of detected passengers who have been assigned the wrong origin or destination. An incorrect O-D might be assigned to the detected passengers because of two reasons. First,

commuters are free to turn on or off the Wi-Fi and Bluetooth modules of their devices during their trips. Therefore, since we estimate the passengers' O-D location based on their received signals, this issue might cause some errors in assigning the origin and destination location. Second, as we explained earlier in Chapter 4, the frequency of transmitting Wi-Fi and Bluetooth signals for mobile devices can be different. Thus, SMATS TrafficBox™ might receive no signal from a mobile device while it is on-board. In this situation, the algorithm considers this passenger as alighted.

For evaluating the results, we compared all the detected O-D data in the estimated matrix with the actual one to find the errors. If a cell value in estimated O-D matrix is more than the actual one, it is as an error; otherwise, if the estimated value is less or equal to the actual one, it is correct. Figure 5.6 illustrates an example of the evaluating process.

	Zone 1	Zone 2	Zone 3
Zone 1	3	4	3
Zone 2	0	2	5
Zone 3	0	0	1

A sample actual O-D matrix

	Zone 1	Zone 2	Zone 3
Zone 1	2	0	4
Zone 2	0	0	3
Zone 3	0	0	3

Error

A sample estimated O-D matrix

Figure 5.6 An example of the evaluating process

After comparing all the values in the original O-D matrix with the estimated one and finding errors, we calculated the Percentage Error by using equation 7.

$$Percentage\ Error = \frac{\sum |Estimated - Actual|}{Total\ detected\ Passengers} \times 100 \qquad (7)$$

In equation 7, the results which are close to zero show that the algorithm estimated the origin and destination of passengers correctly.

5.4.1. Evaluation Results

Table 5.4 demonstrates the trips percent errors for the two proposed approaches (clustering and online threshold-based). The zone errors are calculated by comparing zone-based estimated and real O-D matrices using equation 7.

Table 5.4 Percentage errors in O-D estimation for the two proposed methods (Clustering and Online threshold-based)

	Hierarchical	**K-Means**	**Online**
Trips	**Zone Error (%)**	**Zone Error (%)**	**Zone Error (%)**
First	10.35	10.35	4.25
Second	3.57	3.57	0
Third	4.00	4.00	4.76
Forth	6.06	5.00	0
Fifth	14.81	14.81	6.25
Sixth	2.32	4.54	23.07
Average	**6.85**	**7.045**	**6.38**

The average error for Hierarchical agglomerative, K-Means and online methods are 6.85, 7.045, and 6.38 percent, respectively. It is clear that the online threshold-based method has the lowest percentage of error while the K-Means algorithm has the highest estimating error.

Chapter 6: Conclusions

6.1. Summary of the research

Public transportation companies employ Origin-Destination data for efficient route planning and optimized bus scheduling. Acquiring O-D data using traditional ways is time-consuming and difficult. Therefore, novel approaches are needed to estimate the O-D data using information technologies and data analytics.

Current researches have used different ways for estimating bus passengers' O-D matrix such as Automatic Passenger Count (APC), Automatic Fare Collection (AFC), Automatic Vehicle Location (AVL), and sensing technologies. However, using Wi-Fi and Bluetooth sensors in this application is still new. The main challenge in using Wi-Fi and Bluetooth sensors is distinguishing between passengers and non-passengers' signals as the sensors detect all the transmitted signals from the surrounding environment. To address this issue, previous studies defined various strict thresholds to filter passengers' signals. However, because these thresholds were set in a way to fulfill the requirements of specific scenarios, this approach is not general.

In this research, we proposed two different approaches to estimate bus passengers' origin and destination using their mobile device's Wi-Fi and Bluetooth signals. The SMATS TrafficBox^{TM} sensor collected the surrounding Wi-Fi and Bluetooth MAC addresses, as well as the corresponding GPS data. Then, by joining these data sets, we were able to determine traveler boarding and alighting location.

The idea of the first proposed method was to use clustering algorithms for identifying passengers' signals automatically and without using strict thresholds. For this purpose, the collected data was preprocessed, and five features were extracted from the data set. We then ran K-means and Hierarchical agglomerative clustering algorithms. The experimental results demonstrated that the proposed clustering methods could separate signals inside (passengers) and outside the bus (non-passengers) properly.

The second approach was an online threshold-based analysis of data in which the detected Wi-Fi and Bluetooth signals were analyzed in real time. We proposed four

thresholds to recognize the signals received from the bus passengers properly. This method can help public transit companies to monitor and plan for passengers' flows online.

In the validation phase, we compared all the O-D matrices generated by our two proposed methods with the ground truth data, then calculated each method's accuracy. The final results showed that K-Means, Hierarchical Agglomerative, and threshold-based approaches have 27.14%, 29.39%, and 22.78% detection rate average, respectively, which are promising results compared to the 5% detection rate of survey-based approach. Also, we calculated the zone-based percentage error of each method, and the results showed that the online threshold-based approach with 6.38% has the lowest percentage error average in comparison with Hierarchical and K-Means methods with 6.85% and 7.045% percentage error, respectively.

Although we tested our proposed solutions on a small scale, we believe that using Wi-Fi and Bluetooth signals to monitor passengers' flows and collect information about their Origin-Destination provide useful information about commuters' travel demands. Based on the final results, we infer that Clustering and Threshold-based approaches can distinguish between passengers and non-passengers' signals properly. Also among these methods, it is concluded that Hierarchical Agglomerative clustering has the highest detection rate of passengers while Online threshold-based has the lowest percentage error. Since the percentage error value for our proposed methods are very close, the detection rate values should be considered as well.

6.2. Business Implications of the Research

Traffic congestion is a significant problem in big cities causing delays in everyday commutes. Based on the annual INRIX global traffic report in 2017, direct and indirect costs of congestion for drivers in the USA are more than $305 billion in 2017 alone. The same report also indicates that the time spent stuck in a peak hour traffic could be as long as 102 hours a year which belongs to Los Angles, USA (INRIX, 2017). Thus, there is an obvious for intelligent route planning and scheduling to use resources efficiently and avoid wasting lots of time and money.

Intelligent traffic management using information technologies and data analytics is an essential component of smart cities. This domain uses information technology to improve the quality of life of citizens by enhancing transportation systems and managing traffics. One of the critical elements of smart cities is intelligent public transportation which assures that public transit systems satisfy passengers' demands. Smart public transportation can provide good services for commuters to meet their need in real time and improve their experiences of using public transportation. For enhancing such public transportation services, analyzing operational data and also obtaining passengers' ridership patterns are highly needed. This operating data in smart cities can be obtained by employing sensing technologies on public transit vehicles which facilitate and automate the data collecting procedure. Sensors such as AFC, APC, AVL and Wi-Fi and Bluetooth sensors assist in gathering data in smart public transit.

Machine Learning techniques, such as those proposed in this thesis, facilitate analyzing data collected from the sensors both in real time and offline by using powerful algorithms and methods. The final results can be used for increasing efficiency and effectiveness of public transportation systems, scheduling and route designing, shortening passengers wait time, and also lowering the cost of the public transit companies.

Our industry partner in this project, SMATS Traffic Solutions, is considering integrating our proposed solutions with its analytics software in order to offer novel solutions to its customers for analysis of travel time, origin-destination, as well as queue management. SMATS Traffic Solution is in contact with Ottawa OC-Transpo (Ottawa's public transportation agency) for a long term collaboration to enhance their route planning and scheduling. When this happens, the public will enjoy optimized schedules, with an adequate number of buses deployed on crowded routes, all based on real time analysis of the collected data representing the real-time demands and ridership of citizens. This will undoubtedly enhance the citizens' experiences with using public transportation, with a positive impact on the environment and the quality of life, especially in big cities.

6.3. Limitations and Future Works

As we explained in Chapter 1, the primary objective of this research is to determine bus passengers O-D flows using their mobile devices' Wi-Fi and Bluetooth signals. For achieving this goal, we employed the SMATS TrafficBox^{TM} sensor to capture passengers' signals passively and anonymously, then estimate their movement data. Although the final results are promising, we faced some limitations in this research regarding collecting and analyzing the passengers' signals.

The first limitation is the passengers' use of their mobile devices while they are onboard the bus. Commuters can turn on and off the Wi-Fi or Bluetooth module of their cellphones freely during the trip. As a result, this might cause some errors in determining the origin or destination stops. For example, when a bus passenger turns on his/her device' Wi-Fi or Bluetooth module three stops after getting on the bus, this issue leads to an error in estimating his/her boarding location; because the location corresponding to the first detected signal for each passenger is considered as the boarding location. Similarly, an error might happen when the passenger turns off the Wi-Fi or Bluetooth module several stops before alighting. Additionally, some people may turn off their Wi-Fi or Bluetooth module entirely when in public due to privacy concerns.

In some rare cases, passengers might carry more than one mobile device. In this case, because each device has its own unique MAC address, we might consider these two MAC addresses as different passengers. In this research, we assumed that each passenger carries only one mobile device.

Another limitation is the location of the TrafficBox^{TM} sensor on the bus. Passengers standing on the bus might cause barriers between the TrafficBox^{TM} sensor and onboard mobile devices. Since the sensor was placed on a seat in the middle of the bus, it might not have received some of the Wi-Fi and Bluetooth signals. Therefore, the best solution is to install the TrafficBox^{TM} sensor on the inner ceiling of the bus. Also, more data can be collected from passengers in different hours of the day if we can install the sensor on the bus and monitor commuters flow during each trip. Thus, the final results will not be biased to the specific regions and times, and also the estimated passenger movements become more generalized. For instance, in some hours of a day such as 10:30

AM, we might capture less Wi-Fi and Bluetooth signals because, first, buses are not as crowded as during rush hours and, second, seniors usually commute in this time of the day. Therefore, the possibility that they have smart cellphones or use the Wi-Fi or Bluetooth module while they are onboard is low.

Also, the estimated passengers' flows can be improved by combining sensing technologies with other sources of bus operational data such as APC and AFC technologies, bus door sensors, and bus GPS data (AVL). We only used TrafficBox sensor data in this research.

This study demonstrated that the use of Wi-Fi and Bluetooth technologies to obtain passengers ridership patterns is promising. The potential of proposed methods can be enhanced when this dataset is combined with another source of bus operational data such as AFC and APC. Therefore, using this source of operating data is suggested for future study and improving the accuracy of the final results.

Investigating other features rather than those we employed in this study might result in well-separated clusters and better accuracy. Also, members of each cluster can be labeled as passengers or non-passengers based on the cluster's central point and then the results can be used to train classifiers for distinguishing between passengers and non-passengers' signals.

Moreover, by Installing the TrafficBox^{TM} sensor on the inner ceiling of the bus, more mobile devices signals can be captured during days and months to determine the passengers' patterns unbiased and more precisely.

References

Afshari, H. H., Jalali, S., Ghods, A. H., & Raahemi, B. (2019). An Intelligent Traffic Management System Based on the Wi-Fi and Bluetooth Sensing and Data Clustering. In K. Arai, R. Bhatia, & S. Kapoor (Eds.), *Proceedings of the Future Technologies Conference (FTC) 2018* (pp. 298–312). Cham: Springer International Publishing.

Baeta, N., Fernandes, A., & Ferreira, J. (2016). Tracking Users Mobility at Public Transportation. In *International Conference on Practical Applications of Agents and Multi-Agent Systems* (pp. 224–235). https://doi.org/10.1007/978-3-319-40159-1

Bai, L., Ireson, N., Mazumdar, S., & Ciravegna, F. (2017). Lessons learned using wi-fi and Bluetooth as means to monitor public service usage. *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers on - UbiComp '17*, 432–440. https://doi.org/10.1145/3123024.3124417

Bhaskar, A., Qu, M., Nantes, A., Miska, M., & Chung, E. (2015). Is bus overrepresented in Bluetooth MAC scanner data? Is MAC-ID really unique? *International Journal of Intelligent Transportation Systems Research*, *13*(2), 119–130.

Blogg, M., Semler, C., Hingorani, M., & Troutbeck, R. (2010). Travel time and origin-destination data collection using Bluetooth MAC address readers. In *Australasian transport research forum* (Vol. 36).

Brennan Jr, T. M., Ernst, J. M., Day, C. M., Bullock, D. M., Krogmeier, J. V, & Martchouk, M. (2010). Influence of vertical sensor placement on data collection efficiency from bluetooth MAC address collection devices. *Journal of Transportation Engineering*, *136*(12), 1104–1109.

Canada, S. (2017). Journey to work: Key results from the 2016 Census. Retrieved May 2, 2019, from https://www150.statcan.gc.ca/n1/daily-quotidien/171129/dq171129c-eng.htm

Canon-Lozanol, Y., Melo-Castillo, A., Gomez-Perilla, C. A., Banse, K., & Herrera-

Quintero, L. F. (2013). Web service platform for automatic generation of OlD matrix for mass transportation systems. *13th International Conference on ITS Telecommunications (ITST)*, 462–467.

Choudhary, M. (2018). What is Intelligent Transport System and how it works? Retrieved May 2, 2019, from https://www.geospatialworld.net/blogs/what-is-intelligent-transport-system-and-how-it-works/

Dong, H., & Wang, Y. (2018). Bus Passenger Flow and Running Status Analyzation System Based on MAC Address. In *International Conference on Transportation and Development 2018* (pp. 208–217). https://doi.org/10.1371/journal.pone.0010433

Dunham, M. H. (2006). *Data mining: Introductory and advanced topics*. Pearson Education India.

Dunlap, M., Li, Z., Henrickson, K., & Wang, Y. (2016). Estimation of origin and destination information from bluetooth and Wi-Fi sensing for transit. *Transportation Research Record: Journal of the Transportation Research Board*, (2595), 11–17.

El-Tawab, S., Oram, R., Garcia, M., Johns, C., & Park, B. B. (2017). Data analysis of transit systems using low-cost IoT technology. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on* (pp. 497–502). IEEE.

Elkosantini, S., & Darmoul, S. (2013). Intelligent public transportation systems: A review of architectures and enabling technologies. *2013 International Conference on Advanced Logistics and Transport, ICALT 2013*, 233–238. https://doi.org/10.1109/ICAdLT.2013.6568465

Fukuda, D., Kobayashi, H., Nakanishi, W., Suga, Y., Sriroongvikrai, K., & Choocharukul, K. (2017). Estimation of Paratransit Passenger Boarding/Alighting Locations Using Wi-Fi based Monitoring: Results of Field Testing in Krabi City, Thailand.

Government, U. S. (2016). How GPS works. Retrieved May 2, 2019, from https://www.gps.gov/multimedia/poster/

Government, U. S. (2017). GPS Accuracy. Retrieved May 2, 2019, from

https://www.gps.gov/systems/gps/performance/accuracy/

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *Jouranl of Management Information Systems Quarterly*, *28*(1), 75–105. https://doi.org/10.3969/j.issn.1673-8225.2010.16.023

IBM. (2012). CRISP-DM Help Overview. Retrieved May 2, 2019, from https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm?pos=2

ITSCanada. (n.d.). Overview of ITS. Retrieved May 2, 2019, from https://www.itscanada.ca/education/overview/overview/index.html

Ji, Y., Mishalani, R. G., & McCord, M. R. (2014). Estimating Transit Route OD Flow Matrices from APC Data on Multiple Bus Trips Using the IPF Method with an Iteratively Improved Base: Method and Empirical Evaluation. *Journal of Transportation Engineering*, *140*(5), 04014008. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000647

Ji, Y., Mishalani, R. G., & McCord, M. R. (2015). Transit passenger origin-destination flow estimation: Efficiently combining onboard survey and large automatic passenger count datasets. *Transportation Research Part C: Emerging Technologies*, *58*, 178–192. https://doi.org/10.1016/j.trc.2015.04.021

Ji, Y., Zhao, J., Zhang, Z., & Du, Y. (2017). Estimating Bus Loads and OD Flows Using Location-Stamped Farebox and Wi-Fi Signal Data. *Journal of Advanced Transportation*, *2017*.

Kostakos, V., Camacho, T., & Mantero, C. (2010). Wireless detection of end-to-end passenger trips on public transport buses. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* (pp. 1795–1800). IEEE.

Kostakos, V., Camacho, T., & Mantero, C. (2013). Towards proximity-based passenger sensing on public transport buses. *Personal and Ubiquitous Computing*, *17*(8), 1807–1816.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Leaning, *1*(1), 111–117.

Malinovskiy, Y., Saunier, N., & Wang, Y. (2012). Analysis of Pedestrian Travel with Static Bluetooth Sensors. *Transportation Research Record: Journal of the Transportation Research Board*, *2299*, 137–149. https://doi.org/10.3141/2299-15

McCord, M., Mishalani, R., & Hu, X. (2012). Grouping of Bus Stops for Aggregation of Route-Level Passenger Origin-Destination Flow Matrices. *Transportation Research Record: Journal of the Transportation Research Board*, *2277*, 38–48. https://doi.org/10.3141/2277-05

Mishalani, R. G., McCord, M. R., & Reinhold, T. (2016). Use of Mobile Device Wireless Signals to Determine Transit Route-Level Passenger Origin–Destination Flows. *Transportation Research Record: Journal of the Transportation Research Board*, *2544*, 123–130. https://doi.org/10.3141/2544-14

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Petrovic, S. (2006). A Comparison Between the Silhouette Index and the Davies-Bouldin Index in Labelling IDS Clusters. In *11th Nordic Workshop on Secure IT Systems* (pp. 53–64). https://doi.org/10.1016/S0021-8502(96)00447-8

Shlayan, N., Kurkcu, A., & Ozbay, K. (2016). Exploring pedestrian Bluetooth and WiFi detection at public transportation terminals. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on* (pp. 229–234). IEEE.

SMATS. (n.d.). TrafficBox™. Retrieved May 2, 2019, from https://www.smatstraffic.com/products/trafficbox/

Stallings, W. (2001). Introduction to Bluetooth. Retrieved May 2, 2019, from http://www.informit.com/articles/article.aspx?p=23760

Transportation-Committee. (2011). 2011 O-D Survey. Retrieved May 2, 2019, from http://www.ncr-trans-rcn.ca/surveys/o-d-survey/o-d-survey-2011/

Warren Liao, T. (2005). Clustering of time series data - A survey. *Pattern Recognition*, *38*(11), 1857–1874. https://doi.org/10.1016/j.patcog.2005.01.025

Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *12*(4), 1624–1639. https://doi.org/10.1109/TITS.2011.2158001

Appendix A: Ethics Certificate

The following document is the Ethics certificate of this research.

26/09/2018

Université d'Ottawa
Bureau d'éthique et d'intégrité de la recherche

University of Ottawa
Office of Research Ethics and Integrity

CERTIFICAT D'APPROBATION ÉTHIQUE | CERTIFICATE OF ETHICS APPROVAL

Numéro du dossier / Ethics File Number	H-08-18-809
Titre du projet / Project Title	Estimating Bus Passengers' Origin Destination Travel Route using Data Analytics on Wi-Fi and Bluetooth Signals
Type de projet / Project Type	Thèse de maîtrise / Master's thesis
Statut du projet / Project Status	Approuvé / Approved
Date d'approbation (jj/mm/aaaa) / Approval Date (dd/mm/yyyy)	26/09/2018
Date d'expiration (jj/mm/aaaa) / Expiry Date (dd/mm/yyyy)	25/09/2019

Équipe de recherche / Research Team

Chercheur / Researcher	Affiliation	Role
Shahrzad JALALI	École de science informatique et de génie électrique / School of Electrical Engineering and Computer Science	Chercheur Principal / Principal Investigator
Bijan RAAHEMI	École de gestion Telfer / Telfer School of Management	Superviseur / Supervisor
Hamedhossein AFSHARI	University of Ottawa	Assistant de recherche / Research Assistant

Conditions spéciales ou commentaires / Special conditions or comments

550, rue Cumberland, pièce 154
Ottawa (Ontario) K1N 6N5 Canada

550 Cumberland Street, Room 154
Ottawa, Ontario K1N 6N5 Canada

613-562-5387 • 613-562-5338 • ethique@uOttawa.ca / ethics@uOttawa.ca
www.recherche.uottawa.ca/deontologie | www.recherche.uottawa.ca/ethics

Université d'Ottawa
Bureau d'éthique et d'intégrité de la recherche

University of Ottawa
Office of Research Ethics and Integrity

Le Comité d'éthique de la recherche (CÉR) de l'Université d'Ottawa, opérant conformément à l'*Énoncé de politique des Trois conseils* (2014) et toutes autres lois et tous règlements applicables, a examiné et approuvé la demande d'éthique du projet de recherche ci-nommé.

L'approbation est valide pour la durée indiquée plus haut et est sujette aux conditions énumérées dans la section intitulée "Conditions Spéciales ou Commentaires". Le formulaire « Renouvellement ou Fermeture de Projet » doit être complété quatre semaines avant la date d'échéance indiquée ci-haut afin de demander un renouvellement de cette approbation éthique ou afin de fermer le dossier.

Toutes modifications apportées au projet doivent être approuvées par le CÉR avant leur mise en place, sauf si le participant doit être retiré en raison d'un danger immédiat ou s'il s'agit d'un changement ayant trait à des éléments administratifs ou logistiques du projet. Les chercheurs doivent aviser le CÉR dans les plus brefs délais de tout changement pouvant augmenter le niveau de risque aux participants ou pouvant affecter considérablement le déroulement du projet, rapporter tout évènement imprévu ou indésirable et soumettre toute nouvelle information pouvant nuire à la conduite du projet ou à la sécurité des participants.

The University of Ottawa Research Ethics Board, which operates in accordance with the *Tri-Council Policy Statement* (2014) and other applicable laws and regulations, has examined and approved the ethics application for the above-named research project.

Ethics approval is valid for the period indicated above and is subject to the conditions listed in the section entitled "Special Conditions or Comments". The "Renewal/Project Closure" form must be completedfour weeks before the above-referenced expiry date to request a renewal of this ethics approval or closure of the file.

Any changes made to the project must be approved by the REB before being implemented, except when necessary to remove participants from immediate endangerment or when the modification(s) only pertain to administrative or logistical components of the project. Investigators must also promptly alert the REB of any changes that increase the risk to participant(s), any changes that considerably affect the conduct of the project, all unanticipated and harmful events that occur, and new information that may negatively affect the conduct of the project or the safety of the participant(s).

Germain ZONGO
Responsable d'éthique en recherche / Protocol Officer
Pour/For **Daniel LAGAREC** Président(e) du/ Chair of the **Comité d'éthique de la recherche en sciences sociales et humanités** / **Social Sciences and Humanities Research Ethics Board**

550, rue Cumberland, pièce 154
Ottawa (Ontario) K1N 6N5 Canada

550 Cumberland Street, Room 154
Ottawa, Ontario K1N 6N5 Canada

613-562-5387 • 613-562-5338 • ethique@uOttawa.ca / ethics@uOttawa.ca
www.recherche.uottawa.ca/deontologie | www.recherche.uottawa.ca/ethics

Appendix B: Route 87 bus stops

The following table shows route 87 (Baseline) bus stop names and coordinates.

Stop Sequence	Stop Name	Zone Name	Latitude	Longitude
1	GREENBORO 1A	Hunt Club	45.36009	-75.65880
2	SOUTH KEYS 1D	Hunt Club	45.35313	-75.65500
3	TRANSITWAY / HUNT CLUB	Hunt Club	45.35088	-75.65384
4	HUNT CLUB / PARKWAY	Hunt Club	45.34921	-75.65759
5	DOWNPATRICK / HUNTCLUB	Hunt Club	45.34712	-75.66139
6	DOWNPATRICK / WYMAN	Hunt Club	45.34590	-75.66090
7	DOWNPATRICK / COTTERS	Hunt Club	45.34478	-75.66243
8	DOWNPATRICK / SHANEGAL	Hunt Club	45.34317	-75.66279
9	DOWNPATRICK / UPLANDS	Hunt Club	45.34230	-75.66411
10	UPLANDS / HUNT CLUB	Hunt Club	45.34450	-75.66554
11	UPLANDS / CAHILL	Hunt Club	45.34558	-75.66620
12	CAHILL / UPLANDS	Hunt Club	45.34654	-75.66608
13	CAHILL / TWYFORD	Hunt Club	45.34752	-75.66464
14	CAHILL / MCCARTHY	Hunt Club	45.35035	-75.66435
15	MCCARTHY / PIGEON	Hunt Club	45.35085	-75.66662
16	MCCARTHY / PAUL ANKA	Hunt Club	45.35132	-75.66990
17	PAUL ANKA / MCCARTHY	Hunt Club	45.35153	-75.67216
18	PAUL ANKA / UPLANDS	Hunt Club	45.34945	-75.67469
19	UPLANDS / AD. 3225	Hunt Club	45.34951	-75.67757
20	UPLANDS / HUNTMASTER	Hunt Club	45.34836	-75.68105
21	UPLANDS / HUNTWOOD	Hunt Club	45.34736	-75.68340
22	UPLANDS / AD. 3095	Hunt Club	45.34618	-75.68511
23	UPLANDS / GILLESPIE	Hunt Club	45.34558	-75.68585
24	UPLANDS / ARCHER	Hunt Club	45.34438	-75.68670
25	UPLANDS / BOWESVILLE	Hunt Club	45.34221	-75.69026
26	RIVERSIDE / MALHOTRA	Hunt Club	45.34420	-75.69107
27	RIVERSIDE / RIVERGATE	Hunt Club	45.34677	-75.69196
28	RIVERSIDE / QUESNEL	Alta Vista	45.35420	-75.69227
29	RIVERSIDE / REVELSTOKE	Alta Vista	45.35712	-75.69020
30	RIVERSIDE / AD. 3195	Alta Vista	45.35865	-75.68912
31	RIVERSIDE / WALKLEY	Alta Vista	45.36035	-75.68795
32	RIVERSIDE / MOONEY'S BAY	Alta Vista	45.36537	-75.68873
33	RIDGEWOOD / RIVERSIDE	Alta Vista	45.36754	-75.68908
34	SPRINGLAND / RIDGEWOOD	Alta Vista	45.36854	-75.68586
35	SPRINGLAND / NORBERRY	Alta Vista	45.36996	-75.68288
36	FLANNERY / SPRINGLAND	Alta Vista	45.37057	-75.68174

37	FLANNERY / RAMSGATE	Alta Vista	45.37193	-75.68213
38	FLANNERY / BROOKFIELD	Alta Vista	45.37342	-75.68362
39	BROOKFIELD / HOBSON	Alta Vista	45.37343	-75.68705
40	CANADA POST / BROOKFIELD	Alta Vista	45.37386	-75.68848
41	CANADA POST / HERON	Alta Vista	45.37568	-75.68826
42	HERON AT MOONEY'S BAY STN.	Alta Vista	45.37717	-75.68550
43	HERON 4A	Alta Vista	45.37905	-75.68048
44	DATA CENTRE /BILLINGS BRIDGE	Alta Vista	45.38316	-75.67936
45	BILLINGS BRIDGE 2A	Alta Vista	45.38426	-75.67683
46	PLEASANT PARK 2A	Alta Vista	45.39273	-75.66923
47	RIVERSIDE 2A	Alta Vista	45.39685	-75.66908
48	SMYTH 2A	Alta Vista	45.40117	-75.66659
49	LYCÉE CLAUDEL 2A	Alta Vista	45.40661	-75.66423
50	HURDMAN B	Alta Vista	45.41214	-75.66556
51	LAURIER 2A	Ottawa Center	45.42472	-75.68701
52	MACKENZIE KING 2A	Ottawa Center	45.42413	-75.68999
53	ALBERT / METCALFE	Ottawa Center	45.42182	-75.69619
54	ALBERT / BANK	Ottawa Center	45.42043	-75.69951
55	ALBERT / KENT	Ottawa Center	45.41882	-75.70321
56	ALBERT / BAY	Ottawa Center	45.41711	-75.70708
57	ALBERT / EMPRESS	Inner Ottawa	45.41418	-75.71037
58	LEBRETON 1A	Inner Ottawa	45.41261	-75.71339
59	BAYVIEW 1A	Inner Ottawa	45.40966	-75.72095
60	TUNNEY'S PASTURE D	Ottawa West	45.40367	-75.73729
61	WESTBORO 1B	Ottawa West	45.39646	-75.75250
62	DOMINION 1A	Ottawa West	45.39231	-75.76076
63	WOODROFFE / SAUNDERS	Ottawa West	45.37941	-75.77628
64	WOODROFFE / RICHMOND	Ottawa West	45.37764	-75.77516
65	WOODROFFE / KNIGHTSBRIDGE	Ottawa West	45.37662	-75.77459
66	WOODROFFE / ANTHONY	Ottawa West	45.37548	-75.77385
67	WOODROFFE / FLOWER	Ottawa West	45.37273	-75.77233
68	CARLINGWOOD - LOBLAWS	Ottawa West	45.37215	-75.76799
69	CARLINGWOOD	Ottawa West	45.37140	-75.76920
70	FAIRLAWN / CARLING	Ottawa West	45.36977	-75.77046
71	FAIRLAWN / LENESTER	Ottawa West	45.36678	-75.76915
72	WOODROFFE / GEORGINA	Ottawa West	45.36536	-75.77075
73	WOODROFFE / HIGHWAY 417	Ottawa West	45.36225	-75.76905
74	WOODROFFE / HIGHWAY 417	Merivale	45.35991	-75.76778
75	WOODROFFE / IRIS	Merivale	45.35762	-75.76648
76	WOODROFFE / DOMALATCHY	Merivale	45.35498	-75.76502
77	WOODROFFE / ADIRONDACK	Merivale	45.35320	-75.76403
78	WOODROFFE / BASELINE	Merivale	45.35191	-75.76346
79	BASELINE 1C	Merivale	45.34706	-75.76165

Appendix C: All transit zones in the National Capital Region

The following table indicates different transit zones in the Ottawa-Gatineau region with their zone number.

Zone Number	Zone Name
001	Ottawa Centre
050	Ottawa Inner Area
100	Ottawa East
120	Beacon Hill
140	Alta Vista
180	Hunt Club
200	Merivale
240	Ottawa West
260	Bayshore / Cedarview
300	Orléans
350	Rural East
360	Rural Southeast
400	South Gloucester / Leitrim
425	South Nepean
450	Rural Southwest
500	Kanata / Stittsville
560	Rural West
600	Île de Hull
625	Hull Périphérie
650	Plateau
700	Aylmer
750	Rural Northwest
800	Pointe Gatineau
820	Gatineau Est
840	Rural Northeast
845	Buckingham / Masson-Angers

Appendix D: Route 87 Wait-time plots

The following plots show the wait-time pattern for all the trips.

Secend Trip

Third Trip

Forth Trip

Fifth Trip

Sixth Trip