TRAVEL TIME MEASUREMENT USING MAC ADDRESS-BASED MOBILE DEVICE SENSING TECHNOLOGY: PRINCIPLES, PRACTICE AND CHALLENGES

Mingjian Fu

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Civil Engineering

University of Washington

2018

Committee:

Yinhai Wang, Chair

Jeff Ban

Edward McCormack

Program Authorized to Offer Degree:

Department of Civil and Environmental Engineering

University of Washington

Abstract

TRAVEL TIME MEASUREMENT USING MAC ADDRESS-BASED MOBILE DEVICE SENSING TECHNOLOGY: PRINCIPLES, PRACTICE AND CHALLENGES

Mingjian Fu

Chair of the Supervisory Committee:
Dr. Yinhai Wang
Department of Civil and Environmental Engineering

MAC address-based mobile device sensing is a popular data collection technology widely applied in transportation and other fields. This study focusses on improving existing MAC address-based mobile sensing data travel time measurement algorithm. Several major issues such as MAC address matching due to multiple detections, noises filtering due to different travel modes, and fluctuations due to low samples have been discussed and analyzed based on Acyclica data and License Plate Reader data in Seattle. A density-based clustering outlier filtering algorithm has been applied to improve travel time measurement accuracy. One of the upcoming challenges in MAC

address-based travel time measurement due to MAC address randomization is analyzed, and the potential impact has been investigated by trip sampling rate and MAC collision rate.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Firstly, I would like to express my utmost gratitude to my research supervisor and the director of UW STAR Lab, Dr. Yinhai Wang, for his professional advice, supports, and guidance during my master's study. His professionalism, cutting-edge thinking, self-dedication has inspired me a lot. His encouragement and support have promoted me to go farther than I thought I could go.

Additionally, I am sincerely grateful to Professor Jeff Xuegang Ban and Professor Edward McCormack to serve on my committee. Thanks for reviewing my thesis and giving me valuable suggestions, which has opened my mind and improved my work.

I would also like to express my thanks towards my fellows in STAR Lab. I appreciate the research opportunities and open environment in STAR Lab. A special thanks to Ziyuan Pu for advice and assistance on mobile data collection. Thanks to Zhiyong Cui, Mayuree Binjolkar and visiting professor Xin Fu for sharing ideas on this research topic. Thanks to Ziqiang Zeng, Wenbo Zhu, Ying Jiang and Ye Bi for giving me advices on the paper writing and presentation. Moreover, I wish to thank my fellows who participated in the lab seminar and give me great suggestions.

I would like to express my sincere acknowledgments to anyone who supports me and helps me during my study. Thanks to all the professors and teachers who have taught me. Thanks to all the staffs at Civil and Environment Department for all the help and assistance in the past two years.

Moreover, I would like to give my deepest thanks to my family, my love and my friends. Thank you all for supporting me. I dedicated this thesis to them all.

Chapter 1. INTRODUCTION

1.1 RESEARCH BACKGROUND

Mobile devices, along with the improvement of microchip processing capabilities and the rapid development of high-speed wireless communication standards, have become an indispensable part in people's daily lives. Nowadays mobile devices provide far more than the basic communication, and even change the way people live. The popularity of mobile devices also inspires people trying to collect data from mobile devices. This emerging data collection technology, which called mobile device sensing, has become popular in the past years. Lots of applications across variety of fields include healthcare monitoring[1], environmental analysis[2], and transportation[3], [4], have been adopted mobile phone sensing data as either primary or supplementary data sources.

Currently, there are two major mobile device sensing data collection mechanisms: participatory sensing[5] and non-participatory sensing. Participatory sensing requires device users' action to manually download and start the application. Some applications ask for user's permission to access the embedded onboard sensors such as GPS, the acceleration sensor, microphone, and camera. However, there are difficulties to popularize these applications and get the user involved in the data collection process when the incentive is not enough. As a result, most of the past participatory sensing user studies were small-scale [6]. Unlike the participatory sensing, who aims at deploying mobile devices to form interactive sensor networks, non-participatory sensing only regards the mobile devices as observable and identifiable data samples. Some specific operation of mobile devices, like scanning Wi-Fi network or pairing Bluetooth devices, broadcast the data frames of the device in the surrounding environment. By installing some additional sensors or modules, these data frames could be recognized. The advantage of non-participatory sensing is that data collection process from mobile devices is silent and passive. Moreover, it doesn't need any interactive operation, has no additional costs on the user's side. Comparing to participatory sensing, non-participatory sensing is easier to promote, more flexible and have been widely adopted into different fields.

In the transportation area, non-participatory mobile sensing data mining are currently a hot topic in transportation data analysis, commonly used for multiple purposes such as trip surveys, traffic monitoring, and travel time estimation. Non-participatory mobile sensing data can be categorized into two different types, Cellular based mobile phone data and Wireless media access control address data.

Cellular-based mobile phone data, such as Call Details Record (CDR) data and the sightings data, which are passively generated by the mobile phone when connecting to the cellular towers, can be used to estimate mobile user's location, capture spatial behavior [7], infer transportation mode [8], and simulate traffic flow [9]. The accuracy of the cellular based mobile phone data depends on the density of the cellular towers [10]. However, location uncertainty and oscillation are two of the major issues associated with data. Previous research has concluded that human mobility patterns are overestimated without proper preprocessing methods [11].

Wireless media access control (MAC) address data, which is a part of data from wireless communication frames, requires additional mobile sensors or devices installed in place to capture the data from Wi-Fi and Bluetooth enabled devices. This technique is also called wireless sniffing. MAC address, as an identifier of the network devices acquired from communication frames, can be used for analysis of dynamic of people and vehicle movement [12], [13]. In the previous researches, lots of application have been utilized MAC address data to measuring the travel time with various scenarios, includes motorized vehicles travelling on interstate freeway[14], [15], urban arterial with mixed traffic condition and signalized intersections[16], [17], and non-motorized trail for measuring pedestrians and cyclists travel time [18]. One of the challenges of measuring travel time using MAC address data is the noise which contained into the raw dataset. Proper data processing framework includes filtering techniques, travel time threshold, etc. are needed to apply on the dataset to improve the overall accuracy.

The increasing use of MAC address data also raises user's awareness of the privacy that public concerned that their devices being tracked may lead to privacy issues and potential misuses that can affect the owners of wireless-enabled portable devices [19]. MAC address randomization is a recent strategy implemented by mobile device vendors in order to enhance the privacy and security of mobile devices. By using a randomized MAC address instead of the hardware MAC address, the vendors expect this strategy can avoid devices being tracked directly by their MAC address. In 2014 Apple Inc. first implemented this strategy on iOS 8 and were continuously enhancing it in the following operating systems [20]. Followed by Apple, Android also introduced a MAC address randomization protocol in Android 6.0, although it highly depended on the network card and chipsets, which resulted in various implementation among different device vendors. The mechanism and individual behavior of MAC randomization-enabled devices are still under investigation. Currently, there is no easy way to reveal the real MAC address based on other information in the data frames.

The popularization of MAC address randomization strategy has great impacts on the existing application of mobile sensing technology, since most of them are based on the assumptions that the MAC address is the unique identifier of the devices. The frequently changed MAC address breaks the law and makes the devices untraceable in the system, which may result in the decreasing of the number of valid samples or wrong device matching. Several researchers also mentioned alternative device identification methods such as fingerprinting and using information element frame to identify the devices [21], [22]. However, most of the research papers are only at the device security perspective, there is no research paper focusing on the current and future impacts on the mobile sensing application in the transportation area.

1.2 RESEARCH OBJECTIVE

Inspired by the increasing use of mobile devices sensing and the new challenges led by MAC address randomization, this paper aims at improving existing mobile device sensing travel time measurement methodology to improve the accuracy and evaluating the impact of MAC address randomization. In addition, a mobile sensing traffic data collection and processing framework are proposed to capture wireless signal frames, extract identification information, filtering noise and calculate travel time on urban arterials with mixed transportation modes.

1.3 PROBLEM STATEMENT

In order to measure and estimate travel time on urban arterials using non-participatory MAC address-based mobile sensing technology, several traditional challenges in MAC address matching due to multiple detections, noises filtering due to different travel modes, and fluctuations due to low samples, have been investigated by multiple researchers individually with different processing strategies, in order to pair the MAC address, remove outliers and smooth the travel time. However, there are limited studies comparing the effectiveness of these strategies. In addition, the recent advancement of MAC address randomization which anonymizes mobile devices is one of the most challenging issues that has direct effects on the current trip identification methodology. Currently, most of the data collected by mobile device sensing system have contained randomized MAC addresses. Meanwhile, none of the current studies in transportation mobile sensing application has ever considered and evaluated these new changes. To mitigate the potential impact of MAC address randomization, some studies in cyber security field have raised new ideas such as using information elements and travel patterns as the new identifier of the devices [23], however, it hasn't been applied to transportation area yet. Thus, how to process and utilize the MAC address-based mobile sensing data in mixed-use urban arterial to measure travel time becomes the main problem in this research.

1.4 SCOPE OF THE STUDY

In this work, we reviewed the existing MAC address data processing strategies, evaluated the effectiveness of travel time measurement strategies, investigated the impact of MAC address randomization, and proposed a new framework that integrated the mobile device data collection, detection pairing, and travel time calculation methods. The key contributions of this paper summarized as follow: (1) applying DBSCAN, a density-based clustering algorithm, as a filtering method to separate the motorized-vehicle travel time; (2) evaluating the trend and impact of MAC address randomization on transportation data collection, with the measurement of sampling rates and collision rates.

The rest of the thesis is organized as follows: Chapter 2 is the literature review that review the existing travel time measurement techniques in the past decades and current MAC address based mobile sensing travel time measurement technology. Chapter 3 introduces the existing travel time estimation methodology and applications, including the experimental results using different filtering strategies using mobile device sensing data. Chapter 4 presents a mobile device data sensing and processing framework. Chapter 5 reveals the trend of randomized MAC addresses and evaluates the impact of mobile devices MAC randomization. Chapter 6 concludes the findings in this thesis.

Chapter 2. STATE-OF-ART

2.1 TRAVEL TIME MEASUREMENT TECHNIQUES

In the transportation area, accurate travel time measurement and estimation are very essential. Over the past years, several technologies have been evolved and used for travel time measurement. According to the FHWA Travel Time Data Collection Handbook [24], these travel time measurement technologies can be categorized into three major categories: Test vehicle techniques, license plate matching techniques, and ITS probe vehicle techniques.

2.1.1 *Test vehicle*

Test vehicle techniques, which also called floating car techniques, is based on the exchange of information between a fleet of floating cars traveling on road and a central data system[25], has been used over 40 years as the ground truth for travel time data estimation [14], [26]. Previously the travel time and distance need to be recorded manually by a recorder with pen and stopwatch so that the accuracy highly depends on the trained recorder. Nowadays most of Floating Car Data (FCD) are based on the Global Positioning System (GPS) [27], which is a satellite-based navigation and localization system. Most of the commercial vehicles are equipped with an onboard GPS-based Automatic vehicle location (AVL) system automatically sending a vehicle’s location at an interval to a centralized server. The assistance of the embedded computer or smart devices have provided a convenient way for positioning and have also reduced the cost. Plus, it can provide real-time traffic information with the support of the cellular network, allows a realistic travel time and optimal route calculation for individual and commercial road users [28].

Figure 2.1 INRIX Realtime Traffic Map[29]

Figure 2.1 shows a real-time traffic map from INRIX Inc, whose data comes from highly granular FCD combined with traditional traffic information system[30]. The previous study has utilized buses and trucks equipped with AVL system serve as probes to examine the real-time travel time estimation of general vehicles [31], [32]. The advantage of FCD includes the low cost, data continuity and automated data collection [32]. It is good for all weather and traffic conditions. The limitation of floating car technology is that the accuracy of the travel time depends on the percentage and the geographical coverage of the floating cars traveling on the road. Moreover, AVL system might be failed to receive signals from satellites in tunnels, underpasses, or in a high-rise dense city center [33].

2.1.2 *License plate matching*

Meanwhile, License plate matching is another travel time data collection technique. With the advancement of image recognition and processing technology, the license plate can be recognized and matched by Automatic Number Plate Recognition (ANPR) systems. ANPR system contains travel lane-based video cameras and a backend computer server. This technology has been one of the dominant travel time measurement techniques in the past years, as numerous ANPR systems have been deployed worldwide[34]. Figure 2.2 shows the roadside of video cameras used for the ANPR system in Seattle, WA

Figure 2.2 Video Camera for ANPR System in Seattle, WA

The advantage of license plate matching is it could get a good overall travel time measurement with only small and random samples. The study from the Chicago Area Transportation Study indicates that a minimum 26 license plates reading are essential for a 95 percent confidence level in a given time period [35]. The license plate matching also used for identifying vehicles in origin-destination surveys with several video camera installations. However, one constraint is that the travel time data is limited to locations where observers and video camera are positioned [36]. And the license plate matches the begin timepoint and end time point, only overall travel time can be extracted from these data. Moreover, ANPR system is difficult and could be affected by different illumination factors, vehicle shadows, and non-uniform characters. The accuracy rate for ANPR character recognition rate varies from 85% to 98.7%. None of the existing ANPR algorithms have achieved 100% overall accuracy rate which is also quite not possible [34].

2.1.3 *ITS probe vehicle techniques*

ITS probe vehicle techniques include signpost-based automatic vehicle location technique, automatic vehicle identification (AVI) technique, ground-based radio navigation and cellular geolocation [24].

The signpost-based automatic vehicle location system is based on a beacon installed on the roadside. It has been majorly used by transit agencies to provide real-time transit passenger information, for example, King County Metro in Seattle and Washington and Halifax Transit in Nova Scotia, Canada has utilized this technology on their buses. Some of the signpost-based AVL systems can collect vehicle performance data, such as fuel consumption and oil pressure, and passenger count data, which can be served as a complementary data source for transit operation. The major disadvantage is that signpost-based automatic vehicle location system has a high cost of roadside infrastructure and low accuracy for positioning[37]. It also has a limited coverage restricted by the signpost-based beacon.

Automatic vehicle identification is a technology that use in-vehicle transponders and roadside sensor units to identify vehicles for fleet management reasons [24]. The major purpose of using AVI to identify vehicles is electronic toll collection. With several sensors installed along the road, it is also capable to get the travel time in addition to perform the basic tolling function.

Figure 2.3 Location of E-ZPass Readers in Midtown Manhattan, New York City

A few major cities include New York City, NY, and Houston, TX, have applied AVI technology on traffic monitoring. TRANSCOM's system for managing incidents and traffic (TRANSMIT) is one of the largest AVI-based traffic surveillance system in New Jersey and New York City that depends on E-ZPass electronic toll collection tags [38]. Figure 2.3 shows the location of E-ZPass readers in midtown Manhattan serve as a traffic monitoring sensor[39]. Niver et al have concluded that this AVI system can estimate the link travel time within the 95% confidence level for most cases, offer an opportunity to collect travel time data in real time [40]. AVI technology not only has a relatively higher accuracy with minimal human error but also can detect vehicles travelling at high speed with lane specific information. The limitation of this technology is that it requires electronic tag and infrastructure, as well as a large storage space for data. It may also have clock drift problem if the clocks for all the transponders are not synchronized.

Ground-based radio navigation, which also called radio triangulation, uses the radio signal to navigate by receiving antenna network. The probe vehicles transmit unique vehicle ID via the radio frequency signal to a centralized computer. Like signpost-based automatic vehicle location system, this technology is also majorly used by transit agencies or private companies to manage their fleets. The startup cost is low, and the technology is relatively simple. However, the disadvantage of radio-based location is low accuracy and low penetration rate in urban areas. Nowadays, lots of ground-based applications have been replaced by GPS technologies.

2.2 MAC ADDRESS BASED MOBILE SENSING TECHNOLOGY

2.2.1 *Overview*

With the fast-growing of wireless network telecommunication, the mobile electronic devices, such as mobile phones, smartwatches, tablets, and in-vehicle navigation systems, have become increasing popularly and commonly used, inspires researchers' interest to use the ubiquitous mobile devices as an innovative complementary transportation data source [41]. The development of high-performance computing machines and large-scale distributed databases and systems enables the huge amount of devices-based generated data to be processed and analyzed from milliseconds to seconds, which make the devices-based data more valuable and feasible to traffic real-time analysis. In transportation area, MAC address-based Bluetooth or Wi-Fi detection technology is one of the predominate data collection techniques that serve different purposes, such as travel time measurement, origin-destination survey, and indoor localization and navigation [42]. It also makes a good effort on the pedestrian and cyclist detection, which is quite difficult with traditional methods. In the past years, several applications have been adopted MAC address-based mobile sensing data with different data cleaning processes and algorithms for transportation operation and management, such as location estimation [3], transit waiting time evaluation[43], [44], roadway travel time measurement [14], pedestrians and cyclists monitoring[18], [45], [46], and travel demand forecasting [47]. The process of MAC address matching consists of two steps: data acquisition and MAC address matching [12].

2.2.2 *Data acquisition*

MAC addresses-based data acquisition requires Bluetooth and/or Wi-fi sensors or portable scanners to capture the wireless probe requests or inquire scans to extract useful data. Mobile device sensors installed on the roadside are capable to capture the active devices from vehicles, cyclists, and pedestrians as they passed by. By installing multiple sensors over different places, the travel time, which is one of the most important traffic performance measures[4], can be measured directly based on the time differences between the two sensors. Differ to traditional sensors such as surveillance camera or transponders, mobile device sensors are lightweight and portable, which are good for both short-period data collection and long period traffic monitoring. Normally, three data fields are extracted from the raw data frame: Timestamp, MAC Address, and Received Signal Strength Indicator (RSSI) value. Timestamp records the exact time when devices are detected by the sensor. MAC address, which usually represents in 16 characters divided by the colon, serves as the unique identifier for each network device. Regulated by the Institute of Electrical and Electronics Engineers (IEEE) organization, all the network devices must follow the standard for network communication, including the format of 48-bit MAC address. Subjected to IEEE 802 standard [12], Bluetooth uses an inquiry procedure to discover new devices in the surrounding area to establish connections, while Wi-Fi uses scan procedure to search for the Wi-Fi Access Point (AP). These procedures enable mobile sensors to extract the MAC addresses and other parameters by collecting inquiry data frames in the converge area [49]. RSSI value indicates the signal strength received by the sensor. For the same device and sensor, RSSI value is correlated with the distance between the device and the sensor, as larger RSSI value indicates the mobile device and the sensor are closer. The RSSI value is one of the useful indicators to improve the accuracy of predicted travel time [16].

The amount and quality of MAC address data depend on numerous factors, including hardware and software [41], discovery time and antenna characteristics [18], which is a challenge to calculate accurate travel time. Abedi et al. reviewed and tested the discovery time for both Bluetooth and Wi-Fi devices. According to their findings, the Wi-Fi discovery time was around 1.3s while the Bluetooth was over 10s. The data collection rate for Wi-Fi devices was also 8-10 times higher than Bluetooth devices [13]. Antenna characteristics impact the scanning range of the mobile device sensors. Abedi et al. did an experiment to compare the travel time accuracy between sensors with 2dB and 16dB antenna gain. They concluded that antenna value had a significant effect on MAC address data accuracy. Based on the application scenarios, the small antenna gain is suggested for lower speed travelers such as pedestrians and cyclists, while the bigger value is recommended for fast speed cyclists and vehicles[18]. Plus, MAC address sensors have a designated cycle to receive signals, while the mobile devices also send the probe requests periodically. As a result, the devices may be detected several times or not at all depending on the travel speed and the configuration of the devices. Quayle et al. mentioned installing multiple sensors at a location can increase the likelihood of detecting a device [17].

2.2.3 *MAC addresses matching*

MAC address matching is one of the most important and challenging parts in travel time measurement process. According to the detection range, one MAC address may be detected multiple times at one sensor location. These multiple MAC detections raise a problem of address matching and have a potential impact of the travel time accuracy. Several researchers have used different methods to address this problem. One of the most common strategies is considering the multiple detections as a group, then uses the first or last MAC address in the group. Wang et al. did a field experiment on SR-522 in Seattle to compare the travel time collected by Bluetooth sensors and Automatic License Plate Recognition (ALPR) device, found the last-to-last matching demonstrate better result by minimizing the interaction delay [15]. Tsubota et al also chose the last-to-last strategy for arterial traffic congestion analysis using Bluetooth data, considering of the link travel time and traffic conditions were controlled by the downstream intersection rather than the upstream intersection [50]. Saeedi et al introduced a method to select the timestamp associated with the maximum RSSI value among multiple detections since the RSSI value could reflect the strength of the signal and is correlated with distance, this methodology would minimize the distance between the mobile device and the sensor to avoid the impact of detection range.

2.2.4 *Outlier Filtering*

MAC address based travel time estimation method has been proven capable of providing reliable and high-quality ground truth travel time data on highways in most cases [14]. However, urban arterial road scenario is more complex especially for intersection-to-intersection travel time calculation [50]. During the MAC address matching step, there might be some outliers due to MAC address mismatch with two trips. For example, if sensor 1 detects the MAC address 01:02:03:04:05:06 in the morning, while sensor 2 detects the same MAC address in the afternoon, the travel time would be extremely huge. To address this issue, a travel time threshold based on reasonable speed is required to remove these outliers. Haghani et al assumed the travel time is smoothly transited, as they used a moving average method to flag and remove the outliers for freeway travel time [14]. However, for urban arterials, the traffic signal delay is one of the most important factors contributes to the travel time. Hence, advanced statistical filtering techniques are needed to reduce the impact of the outlier. Quayle et al chose a moving standard deviation method for automatically filtering with the static limit decided by visual inspection [17]. Kieu applied Box-and-Whisker and Median Absolute Deviation to the Bluetooth dataset, and compare the results on weekday and weekend travel patterns [51]. In addition, the mobile sensors cannot distinguish the source of MAC addresses, the data point from non-motor travel mode may also have impacts on the overall accuracy. Yang and Wu implemented a travel mode identification method using a genetic algorithm and machine learning to separate the auto from other travel modes [52], which also effectively improve the travel time measure accuracy for motorized vehicles.

Chapter 3. MAC ADDRESS-BASED TRAVEL TIME MEASUREMENT APPROACH

3.1 OVERVIEW

In this chapter, the effectiveness of existing trip recognition and filtering methods for travel time estimation is evaluated using a mobile sensing dataset and a license plate matching dataset collected in Seattle, WA. A new filtering method based on the density-based clustering method is introduced and compared to other statistical methods.

This chapter is structured as follows: Section 3.2 introduces the MAC address-based Acyclica data and License plate reader data. Section 3.3 presents the roadway scenarios, includes the traffic activities and volumes. The data preparing and travel time calculation process are introduced in Section 3.4, and the comparison between different algorithms and discussions are presented in Section 3.5.

3.2 STUDY DATA

3.2.1 *Overview*

To compare the travel time estimation, two different datasets, License plate matching data and a mobile sensing dataset called Acyclica Go, are selected. These two datasets have similar coverage on Denny Way in Seattle, WA. License plate matching data serves as the ground truth data, while Acyclica Go data are used to estimate the travel time.

3.2.2 *License plate reader data*

License Plate Reader (LPR) data is derived from data collected by roadway surveillance cameras which use optical character recognition on images to read vehicle registration plates. It has proven to be one of the most accurate travel time data collection technology. Based on the data provided by Seattle Department of Transportation (SDOT), 12 different LPR routes which contain multiple segments on arterials in Seattle, WA are installed with license plate readers and surveillance cameras. The locations of surveillance video cameras affiliated to LPR routes are shown in Figure 3.1.

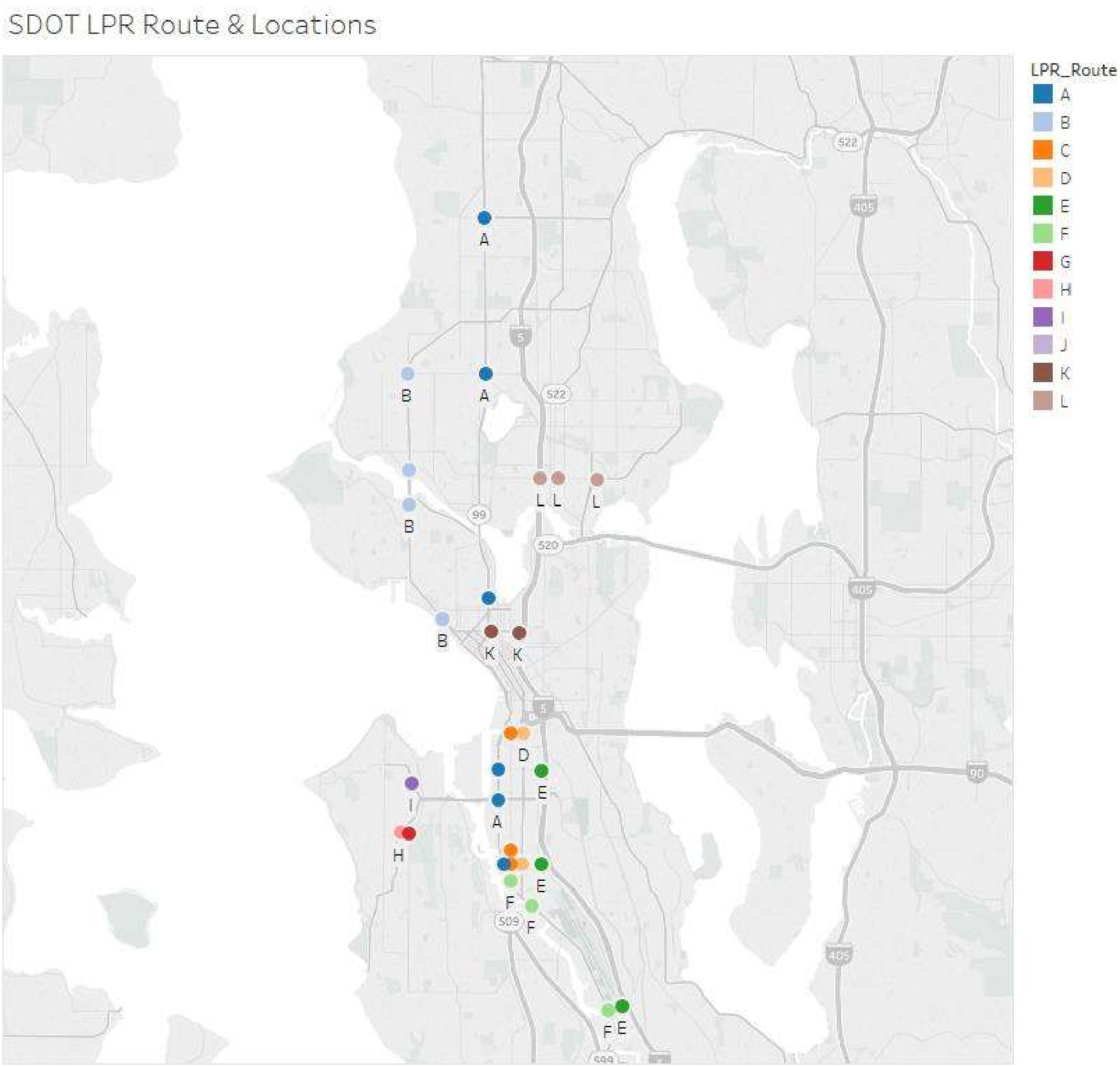

Figure 3.1 LPR Locations and Routes in Seattle, WA

The recorded license plate information can automatically calculate travel time and volume information along each of the segments. According to the LPR dataset description provided by SDOT, this dataset contains segment key, date, time, volume and travel time information for each of the defined segments. Detailed raw data information for each segment is presented in To focus on the travel time data, the data sample of License plate reader data is shown in Table 3.2.

Table 3.1. TravelTime_Seconds is used as the travel time measurement for the further comparison in the following chapters. To focus on the travel time data, the data sample of License plate reader data is shown in Table 3.2.

Table 3.1 LPR Dataset Description

Columns	Data Type	Value Description
SegmentKey	String	Unique ID number assigned to road segments
Date	String	Date of the travel, in the format of "YYYYMMDD"
Time	String	Time of the travel, in the format of "hh:mm:ss"
LaneCount	String	Count of lane number
UpstreamCount	String	Counted vehicle number at the upstream intersection of the segment
DownstreamCount	String	Counted vehicle number at the downstream intersection of the segment
TripsCount	String	Counted vehicle number travel through the whole roadway segment
TravelTime_Seconds	String	Travel time, with the unit of second

Table 3.2 LPR Travel Time Data Sample

SegmentName	DateOfTravel	TimeOfTravel	TravelTime_Seconds	RecordAddDateTime
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:00:00	17	10/9/2014 0:00
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:05:00	16	10/9/2014 0:05
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:10:00	16	10/9/2014 0:10
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:15:00	15	10/9/2014 0:15
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:20:00	0	10/9/2014 0:20
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:25:00	0	10/9/2014 0:25
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:30:00	13	10/9/2014 0:30
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:35:00	16	10/9/2014 0:35
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:40:00	15	10/9/2014 0:40
SEAlrAE031:SEAlrAE041:GPE_Seg	10/9/2014	0:45:00	16	10/9/2014 0:45

3.2.3 *Acyclica data*

Acyclica data is collected by mobile device-based sensors. The sensors can record the MAC address and the Wi-Fi and Bluetooth signal strength of mobile devices or vehicle mounted devices while scanning the Wi-Fi and Bluetooth signals. When these sensors are deployed on multiple intersections, the timestamps from a vehicle or pedestrian recorded by two sensors can be used to estimate the travel time between the two corresponding intersections. According to the Acyclica GO platform, Acyclica sensors have deployed at 100+ intersections in downtown Seattle in late 2018, as shown in Figure 3.2. Raw MAC data could be download from Acyclia Go system and Acyclica application program interface (API). The detailed description of the Acyclica data is shown in Table 3.3, includes timestamp, hashed MAC address and the RSSI strength. Table 3.4 shows the sample of Acyclica data.

Table 3.3 Acyclica Data Description

Columns	**Data Type**	**Value Description**
Timestamp	Timestamp	Date and time of the signal detection
MAC Hash	Varchar	The MAC address of the detection
Strength	Integer	RSSI value, which indicates the strength of the signal

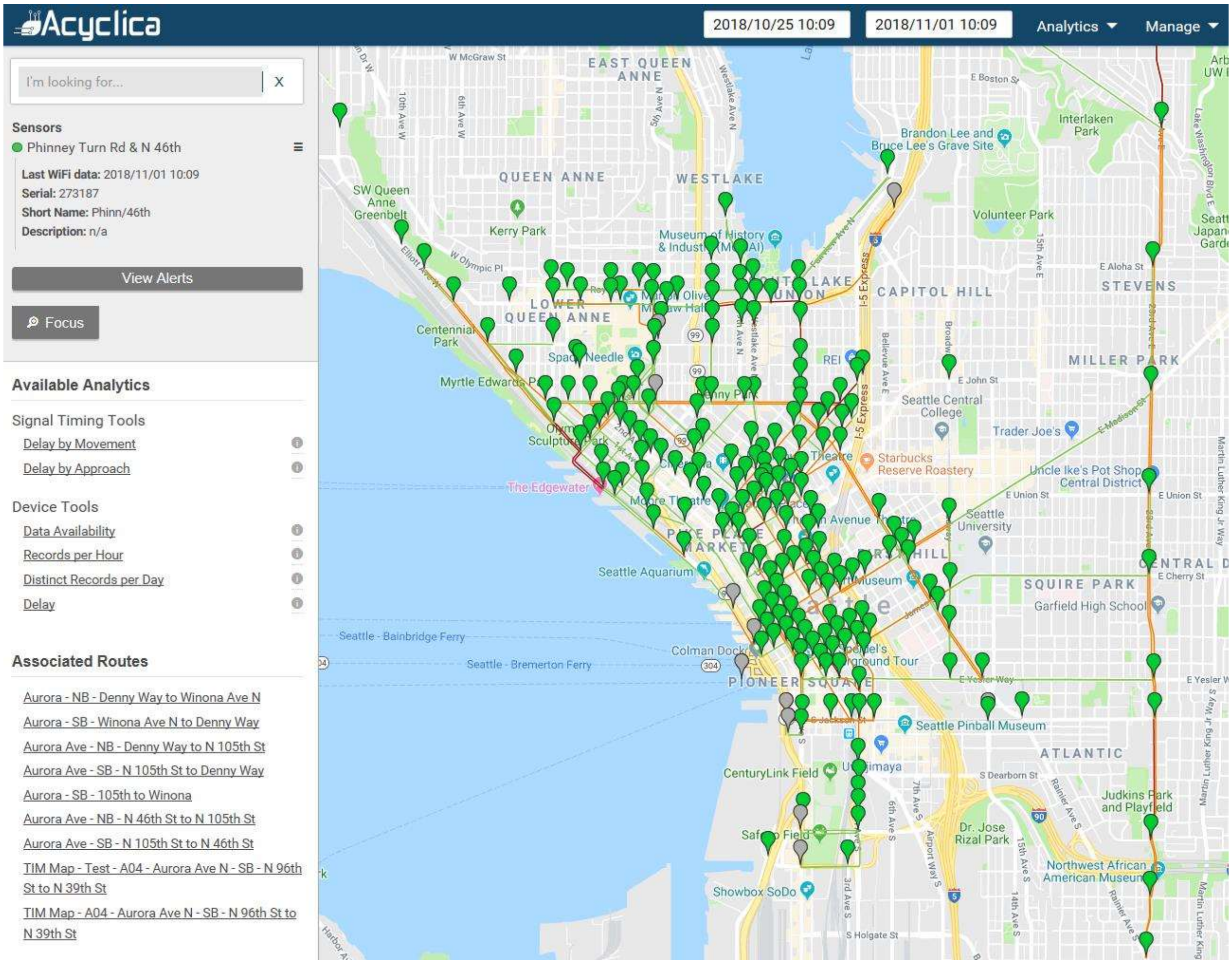

Figure 3.2 Acyclica Go Platform

Table 3.4 Acyclia Data Sample for Sensor No. 273658

Timestamp	MAC Hash	Strength
1427871604.08	47aea082dd61b9f92136a9e65f5e62b77a93c70219354c05a96dfe91883da11c	-62
1427871604.10	47aea082dd61b9f92136a9e65f5e62b77a93c70219354c05a96dfe91883da11c	-62
1427871605.45	273e0c8c9841d5e9acec5a3d7d914cf53b6243c203526e2df548f4059ed863ed	-57
1427871605.63	47aea082dd61b9f92136a9e65f5e62b77a93c70219354c05a96dfe91883da11c	-63
1427871607.30	273e0c8c9841d5e9acec5a3d7d914cf53b6243c203526e2df548f4059ed863ed	-52
1427871608.09	273e0c8c9841d5e9acec5a3d7d914cf53b6243c203526e2df548f4059ed863ed	-51
1427871609.05	e05e6c0c777d8df4339b895688233e97bbb54ac04431000da290bd35aff6a59c	-57
1427871609.13	e05e6c0c777d8df4339b895688233e97bbb54ac04431000da290bd35aff6a59c	-54
1427871609.14	273e0c8c9841d5e9acec5a3d7d914cf53b6243c203526e2df548f4059ed863ed	-53
1427871610.64	af5b5af7348750e76b35aa8dc07f07a0853fd863742865cfb16243fa24297383	-58
1427871613.17	273e0c8c9841d5e9acec5a3d7d914cf53b6243c203526e2df548f4059ed863ed	-56
1427871614.22	273e0c8c9841d5e9acec5a3d7d914cf53b6243c203526e2df548f4059ed863ed	-55
1427871618.75	5fef81682178e14b3cb11c4aff901965540e72e33c8b48408a502bfcfb8bdc5e	-51
1427871621.06	b9767acac04ac03c4ceef2903059f003e02802d072d9bc935e226ac122c1dd5f	-57
1427871621.10	b9767acac04ac03c4ceef2903059f003e02802d072d9bc935e226ac122c1dd5f	-56
1427871621.23	c236a94b2e11acc08daca0f1eb09241ededc5839bb9ca405d89f534991c540fa	-62

3.3 Study Corridor

Based on the data coverage of both LPR data and Acyclica data, the study road segment is decided as Denny Way, which is an important east-west arterial in South Lake Union Neighborhood in Seattle. This route connects the interstate freeway I-5 and SR99, the two major north-south corridors in Seattle metropolitan area. Based on Seattle Department of Transportation (SDOT) Traffic Flow Map Volumes, the average annual weekday traffic (AAWDT) in 2014 various from 22,700 to 23,000 [53]. The speed limit of this road is 35mph. The length between the two intersections is about 2940 ft.

Specifically, the study route is shown in Figure 3.3, between the Dexter Ave N intersection and Stewart Ave/Yale Ave N intersection. This corridor is associated with LPR route K, with 2 license plate readers at each end. It also has several Acyclica sensors along the corridor, which showed in Figure 3.4.

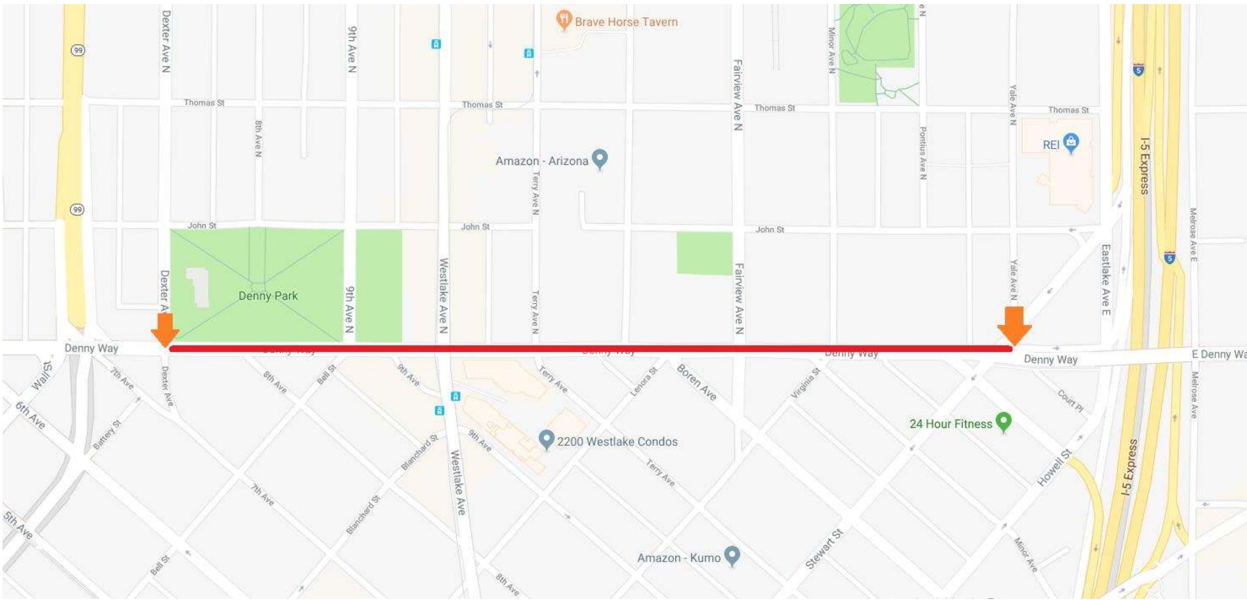

Figure 3.3 Study Corridor and LPR Locations

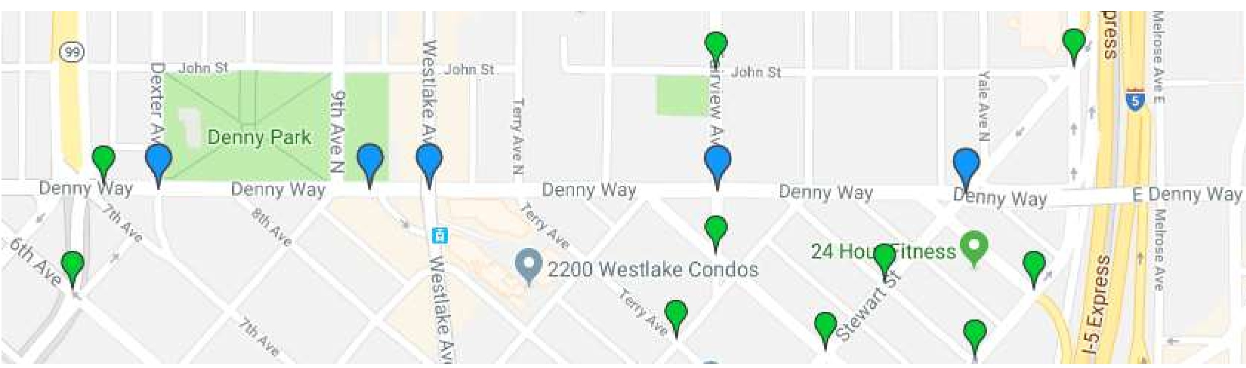

Figure 3.4 Acyclica Sensor Location Near Study Corridor

3.4 METHODOLOGY

3.4.1 *Data Preparing*

Before calculating the travel time for this corridor, the original data sources have been inserted into a PostgreSQL database [54] for persistence and manipulation. The original Acyclica data could be download as comma-separated values (CSV) file for each individual sensor. Two sensors, one at the Dexter Ave and one at the Yale Ave N, are selected to match LPR locations. These CSV files are imported to the database with an additional column called serial represented the ID of the sensor. The timestamp column is converted from integer to formatted datetime string. The LPR data is provided as text files by Seattle Department of Transportation. The LPR travel time, segment description also imports into PostgreSQL database through a similar process.

However, there are some travel time data marked as 0 in the LPR data set due to no vehicles have passed during the 5-minute time window. These data have been removed from the original LPR dataset to smooth the travel time result.

3.4.2 *MAC address matching*

In the Acyclica dataset, as shown in

Table 3.4, a MAC address may be detected multiple times by the same sensor, which causes difficulties in MAC address matching with multiple detections. There are three popular strategies to match the multiple MAC address which leads relatively higher differences in the travel time result. One commonly used strategy is called order-based strategy which uses the first or last detection in a series of detections, is easy to be applied and can measure the intersection delay at one end. The second strategy is selecting the minimum travel time between the two selected sensors. But this strategy is highly depending on the detection ranges, as big antenna gain will result in lower travel time. In addition, previous research has found that the initial discovery time of Bluetooth and Wi-Fi varies a lot [18], which will result in differences between detection modes. Third, RSSI value based-matching strategy targets to find the nearest point between the sensor and detection by using the RSSI value as an indicator. To some extent, it can reduce the uncertainty of the detection time and distance. According to the mechanism of these strategies, RSSI value-based and order-based MAC address matching strategies have different application scenarios. RSSI value can indicate the nearest point which reduces the variation due to detection ranges, is suitable for measuring the travel time with uncontrolled traffic such like the freeway, suburb highway and trails. For signalized intersections and urban arterial, since the delay at the intersection is one of the important factors contribute to the travel time, order-based matching strategy is better for this situation as it only considered the delay from either upstream or downstream intersection.

Based on the Acyclica dataset, there is no indicator to tell the MAC address data is collected from Bluetooth or Wi-Fi. Neither the detection range nor antenna gain is clear. Therefore, to minimize the matching error, the last-to-last matching strategy is selected to calculate the travel time.

According to the LPR data, as the ground truth travel time between these two selected intersections varies from 50 seconds to about 700 seconds, a maximum allowance gap between two detections is set to 800 seconds. If the two detections associated with the same MAC address and same sensor are less than the maximum allowance gap, it would be considered as one multiple detection groups.

3.4.3 *Travel Time Outlier Filtering*

After the MAC matching step, the raw travel time from MAC address have been calculated based on the corresponding MAC address data. However, the raw travel time contains lots of noise due to non-motorized travel patterns, stopovers, and other travel routes. Travel time outlier filtering is an essential part to remove the outliers from the raw result to improve the overall accuracy of the travel time calculation. In the past researches, various outlier filtering algorithms have been applied to the MAC address-based travel time dataset. In this research, three different algorithms, Moving Median Filter, Moving Absolute Deviation (MAD) and Box-and-Whisker are applied to the dataset. The effectiveness of filtering outlier for each algorithm has been visualized and compared.

3.4.3.1 Moving Median Filtering

Median filtering is a useful technique for noise suppression especially suggested for time series data [55]. Wang et al. have applied this technique to the Bluetooth based travel time dataset collected from SR-522 field experiment to remove outliers[15]. By applying a fixed length time window moving around the data points, the moving median filtering algorithm mostly focusses on removing the spikes in a short time period. One advantage is that this algorithm can keep the original sharp edges, which is very essential for peak hour travel time estimation. With the time window setting as *L* seconds, if the target travel time data is at timestamp *Ti*, a group of travel time data entries is set from $Ti - \frac{L}{2}$ to $Ti + \frac{L}{2}$.

To apply this filter, every data point will be screened once to determine whether is accepted or rejected according to the upper bound and the lower bound. The upper bound is calculated as Equation (1) by median value and standard deviation. Generally, the lower bound is the differences between the median value and standard deviation. However, in arterial travel time calculation scenario, motorized vehicles are already the fastest travel mode on the road. The lower bound of travel time would be set based on the travel time calculated by a reasonable free flow speed. As the speed limit is around 30-35mph, the lower bound is calculated as Equation (2) with the reasonable free flow speed set as Speed_Limit+30. Plus, the time window is set as 15 minutes, as the length of 3 bins. If the travel time exceeds the range between the upper bound and the lower bound, it would be considered as an outlier.

$$UpperBound = \sigma + Median(X) \tag{1}$$

$$LowerBound = \frac{Distance}{(Speed_{limit}+30)} \tag{2}$$

3.4.3.2 Moving Median Absolute Deviation

Moving Median Absolute Deviation is another popular technique to find the potential outlier. Like the Moving Median Filtering, Moving Absolute Deviation also requires a sliding time window. Median Absolute Deviation (MAD) is defined as the median of the deviation, which is shown in the Equation (3). The desire upper bound is defined as the sum of MAD and the median value. The algorithm of Moving Absolute Deviation is shown below:

$$MAD = 1.428 \times Median(|X_i - Median(X)|) \tag{3}$$

$$UpperBound = MAD + Median(X) \tag{4}$$

$$LowerBound = \frac{Distance}{(Speed_{limit}+30)} \tag{5}$$

In the Equation (3), (4), X is the travel time data list in the time window, X_i represents the i-th travel time in the list. As mentioned in Section 3.4.3.2, the lower bound is set based on the distance and the maximum desired speed.

3.4.3.3 Box-and-whisker Filtering

The concept of Box-and-whisker filtering methods is similar to the idea of box plot in the statistical area, which is mainly based on the terminology median and quartile. The lower quartile (LQ) represents the travel time at 25th percentile, while the upper quantile (UQ) represents 75th percentile travel time. The interquartile range (IQR) is defined in the Equation (6) as the differences between the LQ and UQ, and the formula of upper and lower bound is defined as Formula (7), (8). As mentioned in Section 3.4.3.2, the lower bound is set based on the distance and the maximum desired speed.

$$IQR = UQ - LQ \tag{6}$$

$$UpperBound = UQ + 1.5 \times IQR \tag{7}$$

$$LowerBound = \frac{Distance}{(Speed_{limit}+30)} \tag{8}$$

3.4.3.4 DBSCAN Clustering

Density-based spatial clustering of application with noise (DBSCAN) is a popular non-supervised clustering algorithm in machine learning and data mining [56]. It relies on a density of the data points that can cluster data in high density. The advantage of DBSCAN is it can effectively discover clusters of arbitrary shape and resistant to noises, which is difficult for centroid-based clustering algorithms such as K-means. There are two major criteria for DBSCAN algorithm: Maximum radius ε and minimum number of points *MinPt*. The $\varepsilon - \text{Neighborhood}$ is defined in Equation (9).

$$N_{\varepsilon}(p) := \{\mathrm{q} \in \mathrm{D} \mid dist(p,q) < \varepsilon\} \tag{9}$$

The $N_{\varepsilon}(p)$ represents the neighborhood of a point in distance ε. $dist(p,q)$ is the distance between two data points, p and q, to measure the similarity of the data points. Each point is categorized as core point, border point and noise point based on the density. A cluster is decided with satisfying density-reachable and density-connected criteria with at least *MinPt* points. However, determining the proper value of maximum radius ε and minimum number of points *MinPt* is difficult, as the clustering result is quite sensitive to these parameters.

Since the travel time data collected by MAC address contain lots of noise, and the trend of travel time is not a regular shape. The DBSCAN clustering algorithm could be applied to filter out the noise.

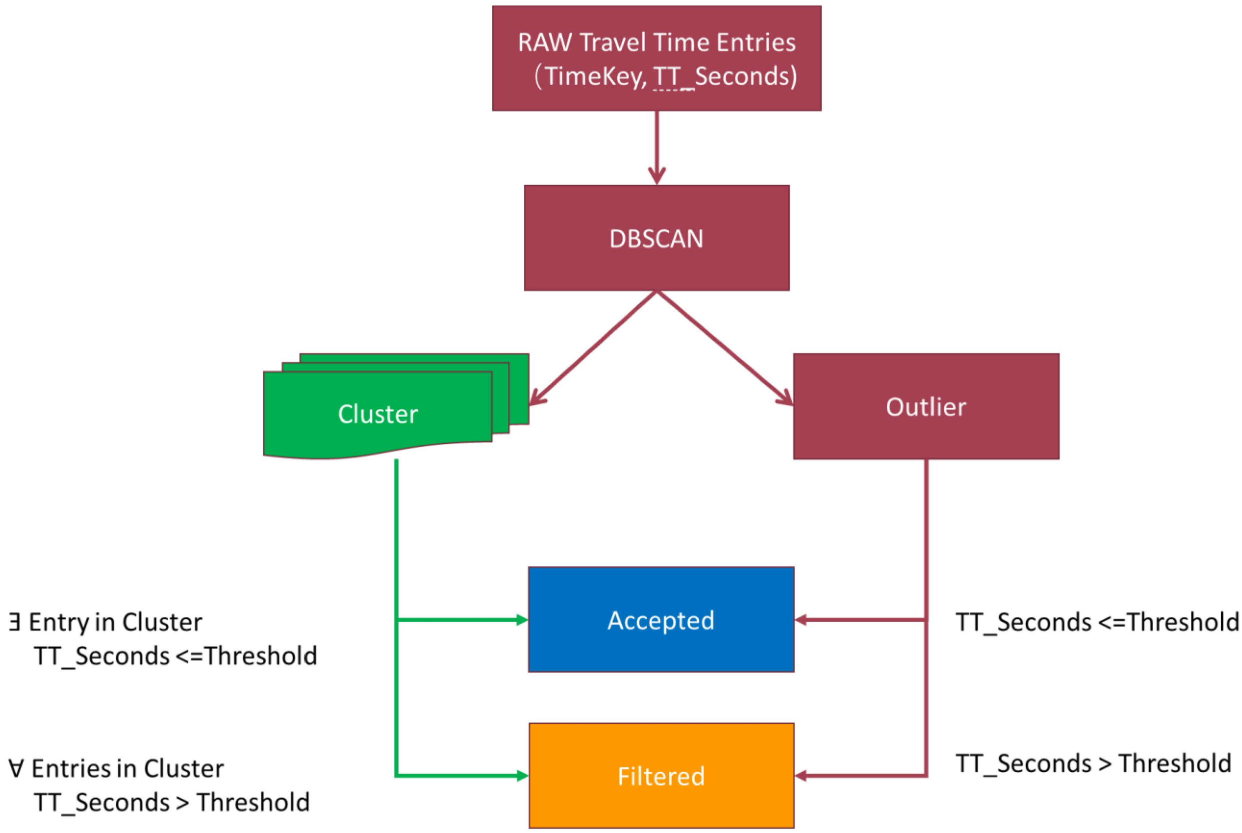

Figure 3.5 The Process of DBSCAN Travel Time Filtering

However, according to DBSCAN algorithm, the result might contain multiple clusters, a union strategy has been developed to combine multiple clusters to reach the result. In addition, some travel time data point might be marked as outlier since the traffic volume around midnight may not be large enough to meet the requirement of *MinPt*. To mitigate this issue, all the travel time less than the Valid Travel Time Threshold would be considered as valid data points. Figure 3.5 shows the overall process for travel time filtering using DBSCAN algorithm. The Valid Travel Time Threshold is determined based on the average travel time on the road corridor and visual inspection, which is set as 150 seconds for the road scenario.

3.4.4 *Time aggregation*

Time aggregation is used to summarize the travel time and make the MAC-address based travel time result comparable to the ground truth data. Since the travel time may start and end at any moment, time aggregation is one essential step to combine the travel time result. As LPR data provides 5-minute aggregate mean travel time, the MAC-addresses based travel time follows the same time resolution. Firstly, each day is divided into 288 exclusive 5-minunte bins, which called *TIME_KEY*, range from 0 to 287. *TIME_KEY*=0 relates to the time at 00:00 AM, and *TIME_KEY*=0 relates to 00:05 AM, etc. Each the MAC-address based travel time data entry is placed into the corresponding bin as the timestamp converted to *TIME_KEY*.

3.4.5 *Travel Time Correlation Analysis*

Travel time correlation analysis aims at measuring the similarity of two series of travel time from MAC-address based and the ground truth data. Pearson correlation coefficient [57], which is a measurement to find the linear relation between two series of data, is used to compare these two travel time variables. The result ranges from -1 to +1, where 1 is total positive linear correlation and 0 is no linear correlation. If the value approaches 1, it means these two variables follow the same trend. The equation of the Pearson correlation coefficient is presented in the following equation:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{10}$$

where $cov(X,Y)$ is the covariance of the series X and Y. σ_X and σ_Y are the standard deviation of X and Y.

3.5 RESULT AND DISCUSSION

3.5.1 *MAC address matching*

As the first step of the travel time calculation, MAC address matching strategies are a direct impact on the accuracy of the overall result. In this experiment, the raw travel time result is calculated based on RSSI value matching strategy, which is using the detection with the maximum RSSI value as the desired start or end point if the MAC address is detected multiple times by the same sensor.

Figure 3.6 shows travel time result calculated from paired MAC address data on April 8, 2015, a normal weekday, and Figure 3.8 shows the travel time result on weekends, April 12, 2015. In these figures, X-axis is the time of the day that represented by TIME_KEY, ranging from 0 to 287. Y-axis is the travel time in seconds. Each blue point is one travel time data matched by MAC address. The minimum travel time at around 50 seconds, which represents the free flow speed at around 40mph as the length between the two intersections is 2940ft. Although most of the points are gathered at the bottom of the chart around 150-200s with reasonable travel time, there are still lots of the points of noise range from 300s to 800s. It might be due to stop-and-go scenarios, different travel routes, or different travel modes such as transit or walking.

Figure 3.7 and Figure 3.9 show the ground truth travel time which is extracted from LPR dataset. During some time period, there might be no vehicles passed through, which result in the travel time as 0 in the raw data. The invalid data have been removed from the dataset. On weekdays, two obvious peaks are observed at around 9:30 and 17:30. On weekends, the travel time is approximate to uniform distributed. Compared to the LPR travel time in Figure 3.7 and Figure 3.9. the same traffic patterns can be roughly observed in both Figure 3.6 and Figure 3.8. However, outliers in the figures prevent estimating the travel time accurately. Appropriate filters need to be applied to remove the outliers.

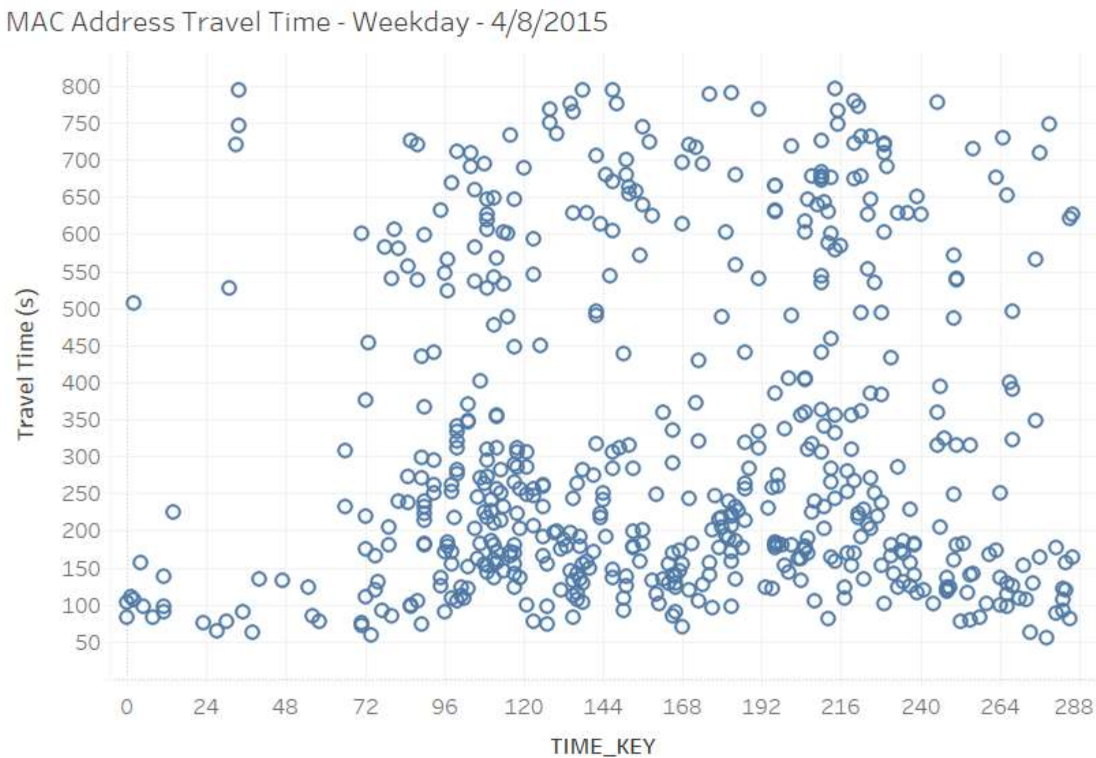

Figure 3.6 Weekday Travel Time Calculated by MAC address

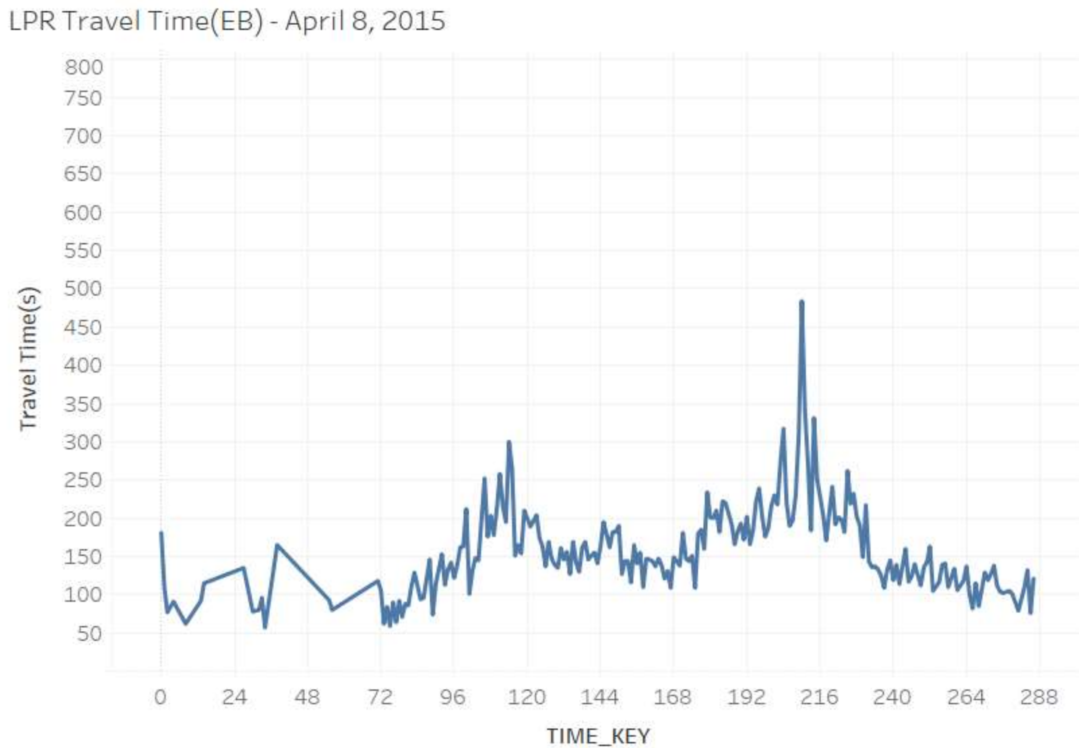

Figure 3.7 Weekday Travel Time Calculated by LPR

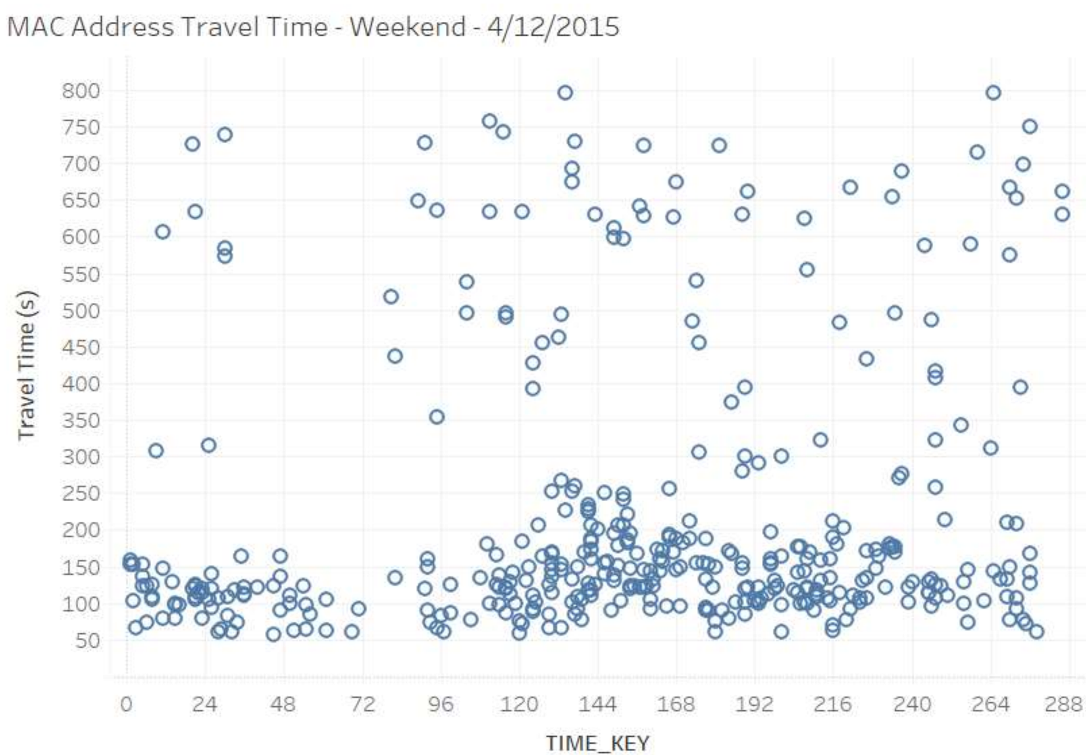

Figure 3.8 Weekend Travel Time Calculated by MAC address

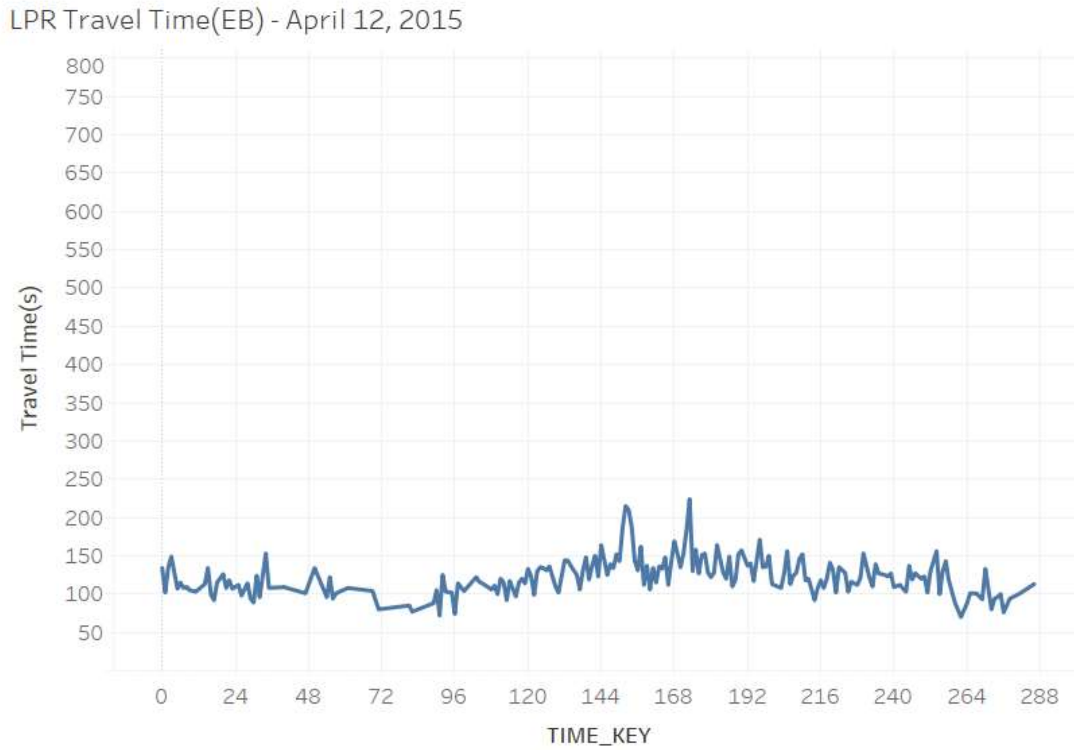

Figure 3.9 Weekend Travel Time Calculated by LPR

3.5.2 *Travel Time Outlier Filtering*

Figure 3.10 shows the overall result of applying different filters introduced in Section 3.5.2 that are applied to the raw result to remove the noise to the travel time dataset. The DBSCAN has the highest filtering rates which remove about a quarter of data, and the MAD and Moving Median Filtering have the similar overall filtering rates. The Box-And-Whisker is the lowest one which only removes 6.77% of total data. Figure 3.11 and Figure 3.12 show the detailed result of the effectiveness of different filtering algorithms on weekday and weekends.

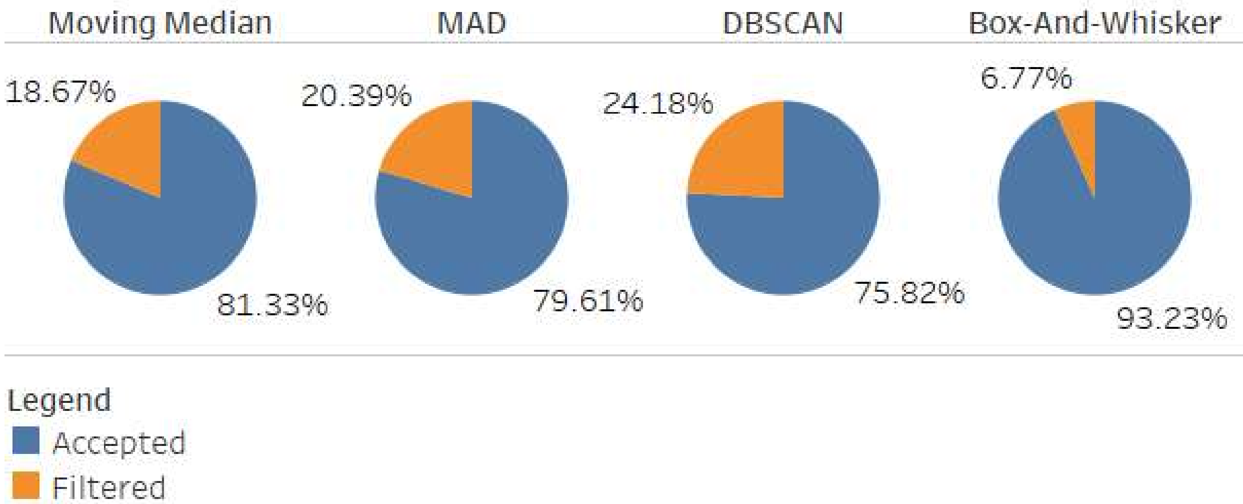

Figure 3.10 Compare Overall Filtering Rates among Different Filters

Moving Median Filter removes the outlier based on the mean and standard deviation. If a particular travel time measurement is within one standard deviation above the localized mean, it is accepted as a valid data point [15]. According to the figures, Moving Median filter can effectively remove noises above 400s. MAD Filtering, which is based on the median and the MAD, has similar effects to the Moving Median Filter. However, these two filtering methods have a limitation to capturing the travel time peaks. DBSCAN which is a density-based clustering method does a better job as two travel time peaks on weekdays is observed in Figure 3.11. Compared to the other three algorithms, the threshold of Box-And-Whisker algorithm is much higher, although it can remove the outliers to some extent, lots of noise points still remain in Figure 3.11. On weekends, although all these filter algorithms have similar results for travel time filtering, DBSCAN tends out to be the most accurate algorithm according to Figure 3.12.

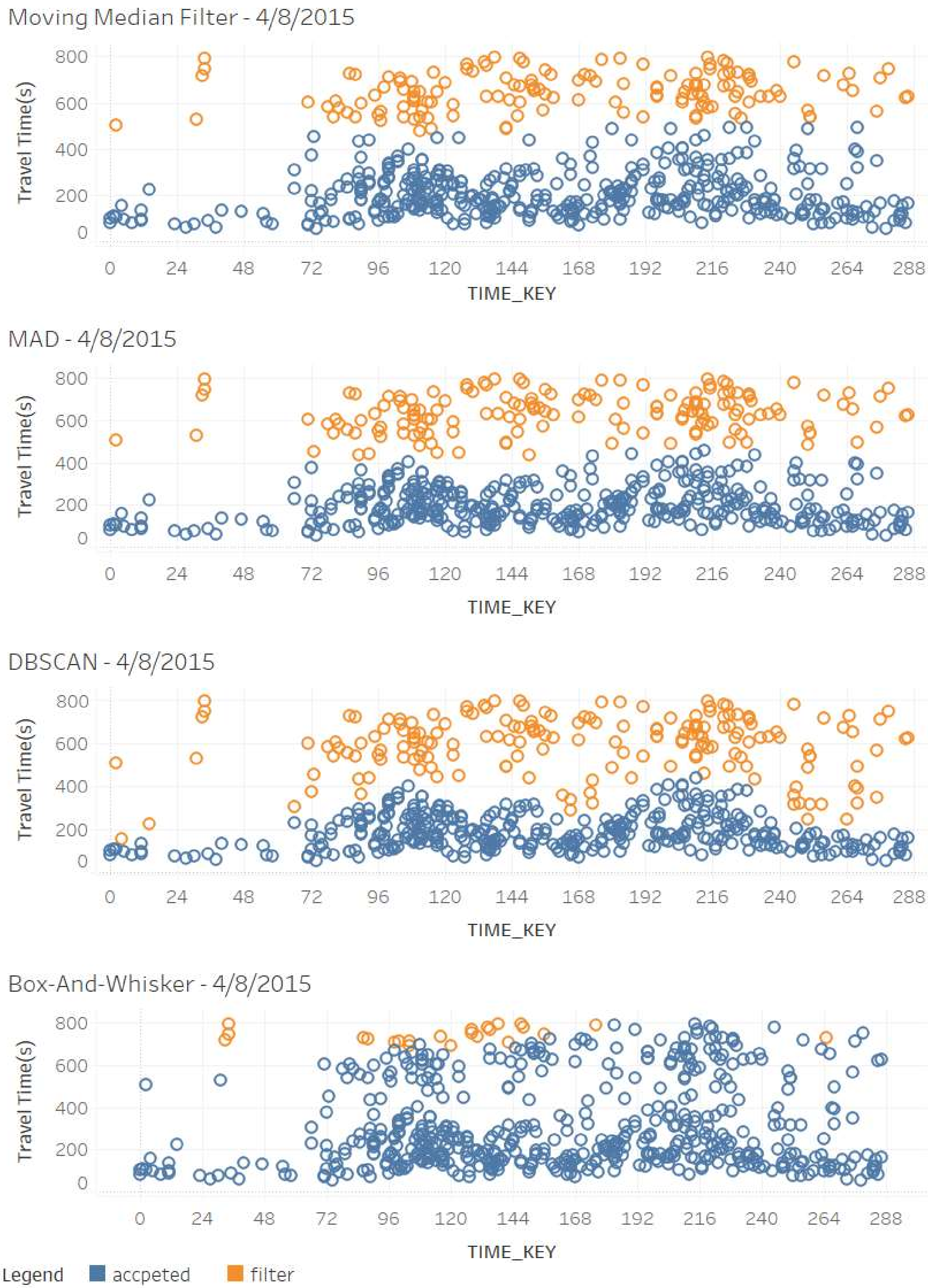

Figure 3.11 Weekday Travel Time Result of Different Filter

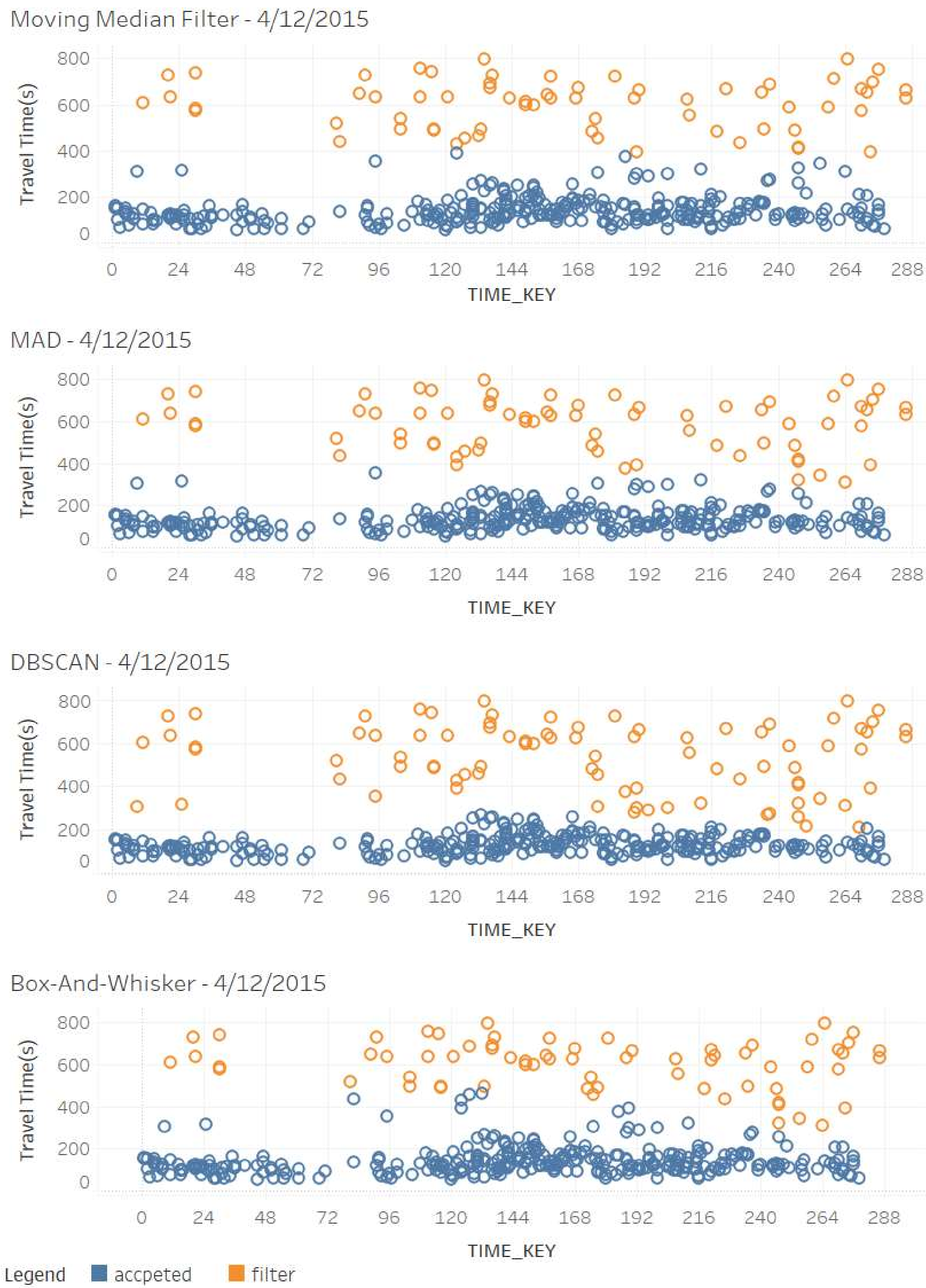

Figure 3.12 Weekend Travel Time Result of Different Filter

3.5.3 *Travel Time Comparison*

3.5.3.1 Travel time comparison between LPR and Acyclica data

Figure 3.13 and Figure 3.14 show the comparison between the travel time calculated by MAC address and LPR respectively on weekday and weekend by using different selection strategy. Each figure includes the travel time result of applying different filters and the raw data. As the raw MAC address travel time results have lots of fluctuations, the filtered travel time is more smooth and close to the ground truth travel time calculated by LPR data.

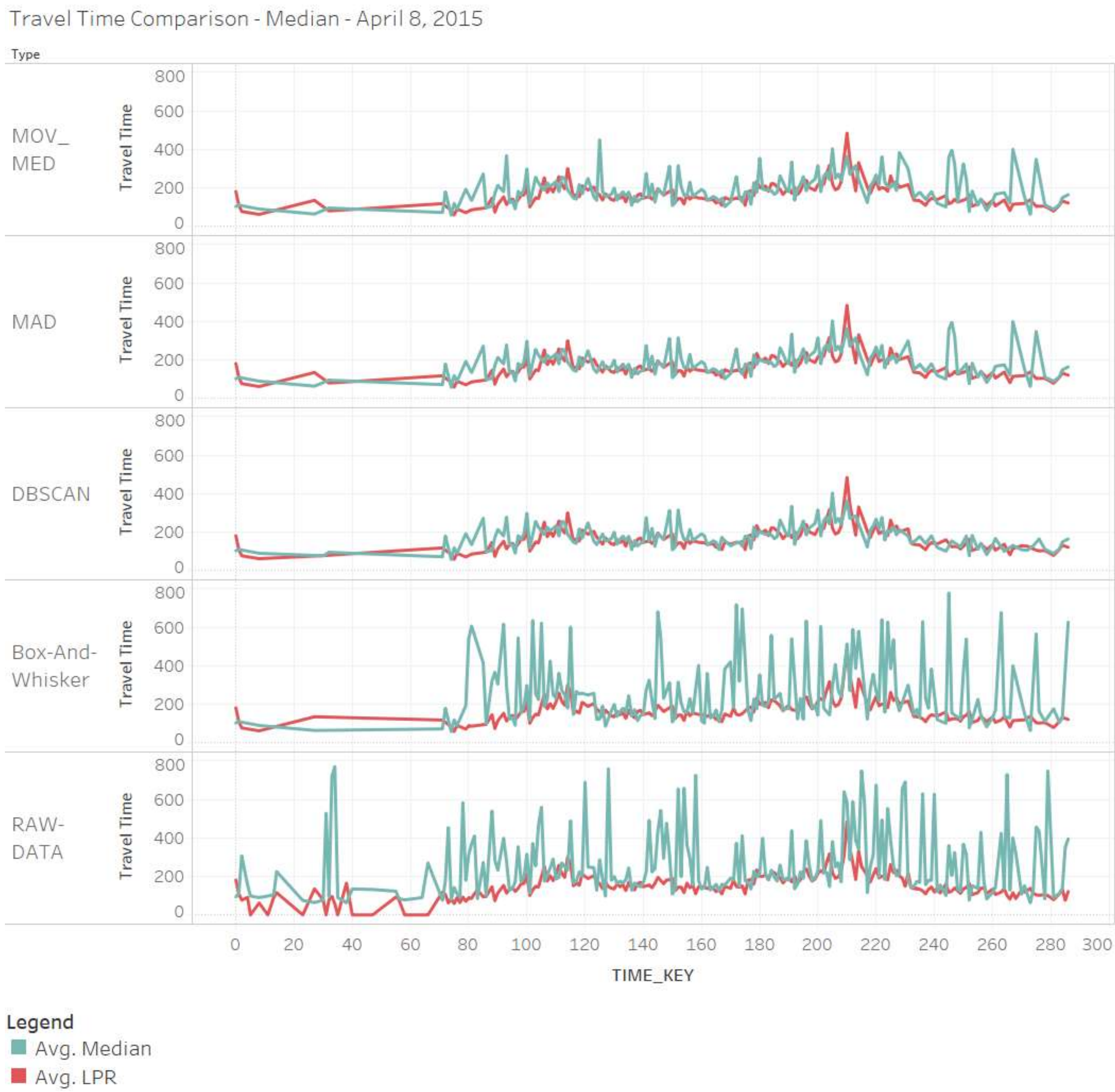

Figure 3.13 Weekday Travel Time Result Comparison: LPR vs Median-MAC

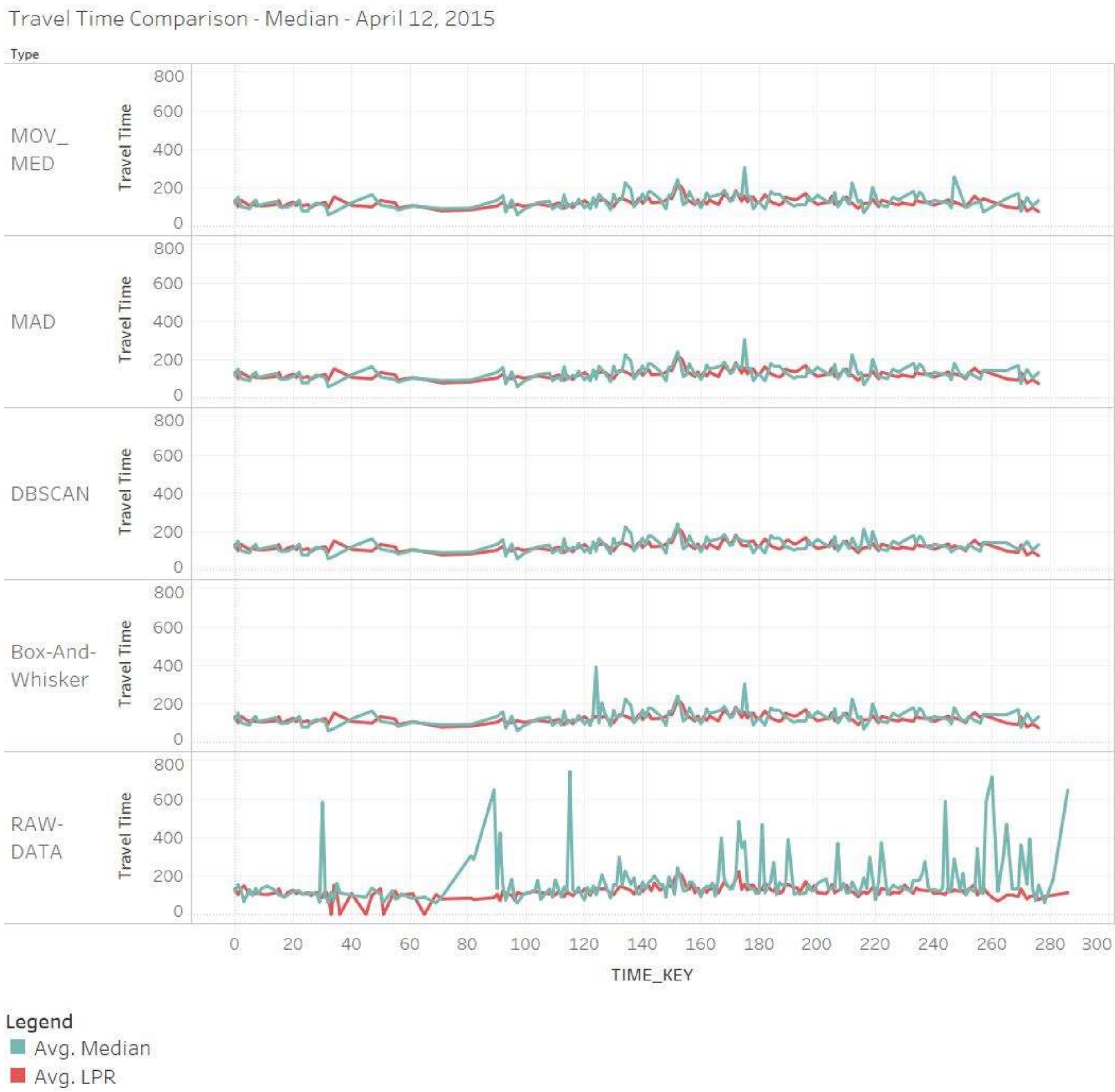

Figure 3.14 Weekend Travel Time Result Comparison: LPR vs Median-MAC

Table 3.5 Travel Time Deviation by Day of Week

Row Labels	DBSCAN	MAD	MOV_MED	Box-And-Whisker
Weekday Avg	**47.14**	**54.02**	**58.30**	**114.36**
Mon	42.79	52.17	57.11	89.54
Tue	46.99	51.77	56.60	117.89
Wed	49.73	54.97	58.55	118.66
Thu	50.23	54.40	58.22	133.54
Fri	45.65	56.32	60.66	109.26
Weekend Avg	**33.57**	**38.71**	**41.32**	**55.17**
Sun	31.58	35.80	38.38	41.55
Sat	35.27	41.21	43.84	66.67
Average	**43.11**	**49.41**	**53.20**	**97.17**

Table 3.5 shows the absolute value of travel time deviation group by day of the week. The average travel time on weekdays is larger than the difference on weekends. One of the reasons is that the traffic is flatter on weekends while having no obvious peaks or valleys. Among the four different filtering algorithms, DBSCAN, MAD and Moving Median algorithms have reached a comparable accurate result, while the travel time result after applying Box-and-Whisker is not very accurate.

3.5.3.2 Travel Time Accuracy

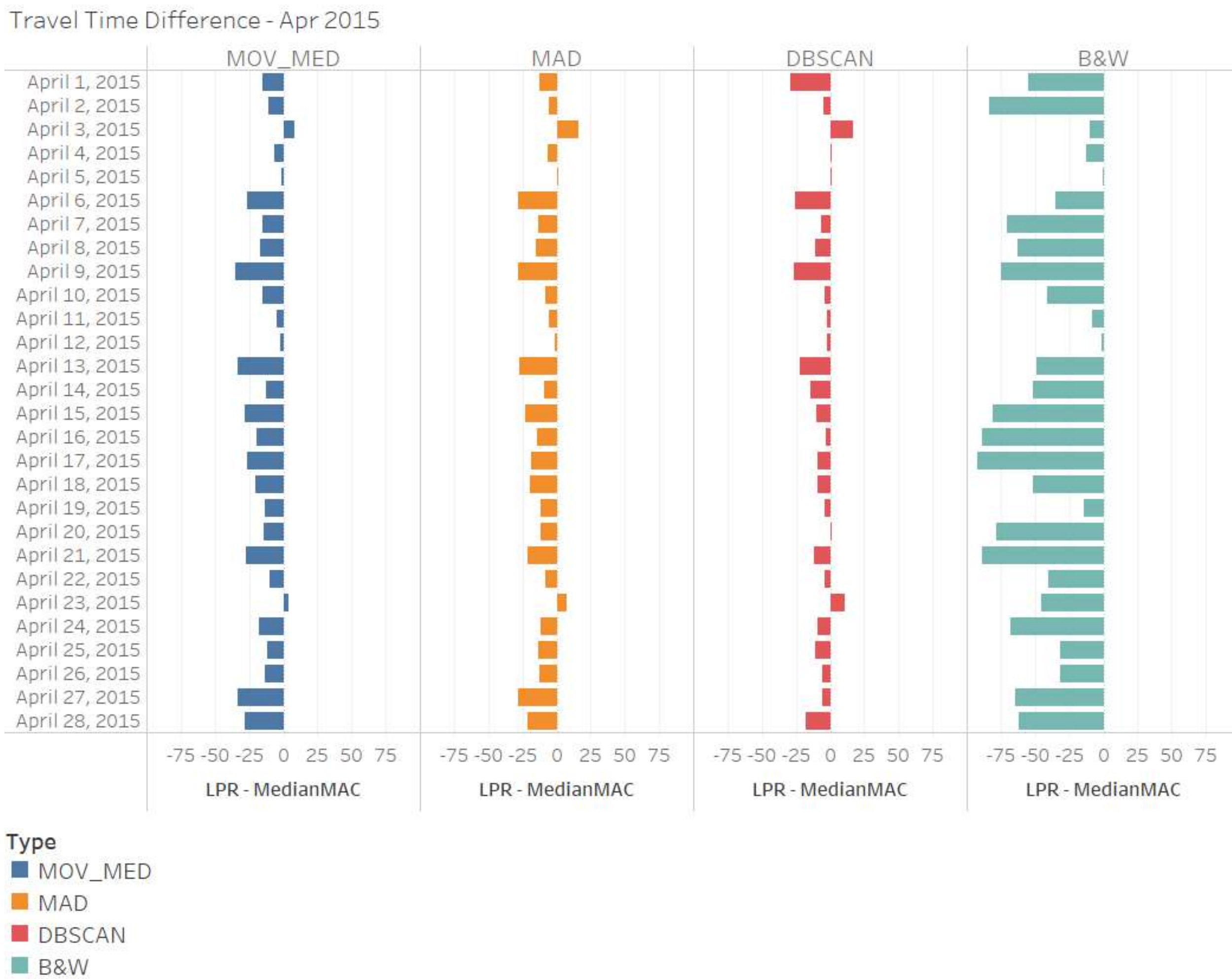

Figure 3.15 Daily Travel Time Difference: LPR vs Median-MAC

Figure 3.15 shows the travel time differences between LPR and MAC Address by using median value to estimate the travel time. Among different filters, Box-and-Whisker has the highest travel time differences ranging from -50 to 0 seconds, which indicates that the travel time calculated by MAC address is higher than the ground truth. Moving Median, DBSCAN and MAD have similar travel time differences around -25 seconds, and DBSCAN reaches the smallest differences among the 4 filtering algorithms.

Figure 3.16 shows the travel time difference histogram which compares the accuracy of MAC-address based travel time results among different filters. The Y-axis represents the percentage of the total number of travel time entries, and the X-axis represents the absolute value of MAC address-based travel time deviation from the LPR travel time. The width of each bin is 25 seconds.

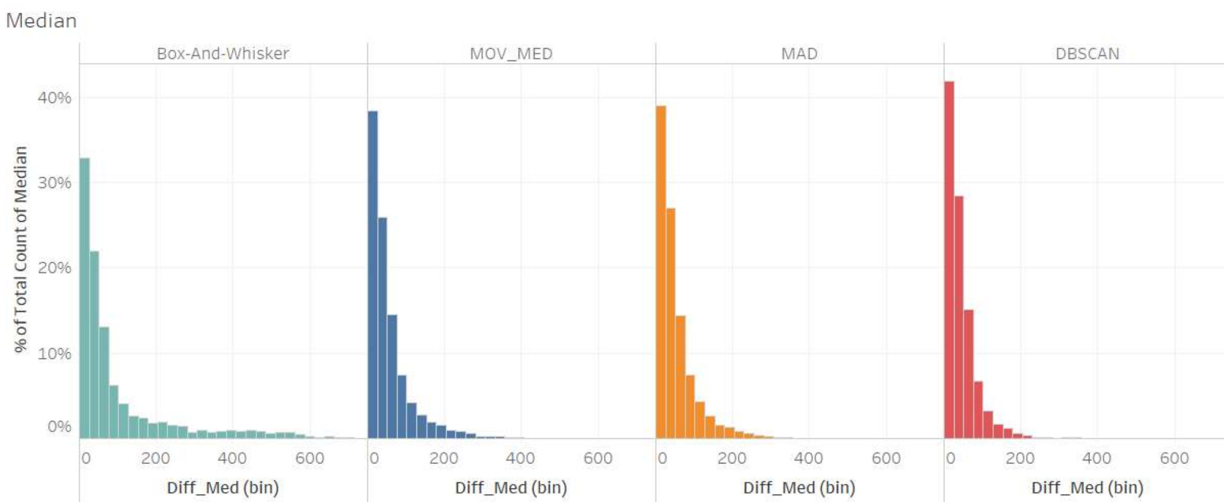

Figure 3.16 Travel Time Difference Histogram

3.5.3.3 *Sample Size*

Figure 3.17 is a box plot that compares the travel time deviation versus the number of samples with different filtering algorithms. The travel time deviation is the differences in seconds between 5-minute aggregated travel time calculated by LPR and the median value of travel time in a 5-minute time period calculated by MAC address. The number of samples indicates how many samples are used to calculate the median value in a 5-minute time period, ranging from 1 to 8. For all the filtering algorithms, there are similar trends that the fluctuation is lower as the number of samples gets larger. As more travel time samples are collected from MAC address in a fixed 5-minutes time period, the overall travel time result is more reliable. Besides, in Figure 3.17, the travel time difference between LPR and MAC address is less than 0, indicating the travel time estimated by MAC address median value is overestimated. One of the reasons is that the installation location of license plate reader and mobile device sensor may not be exactly the same location as LPR. Detection ranges, clock differences, and other system error factors may also contribute to this result.

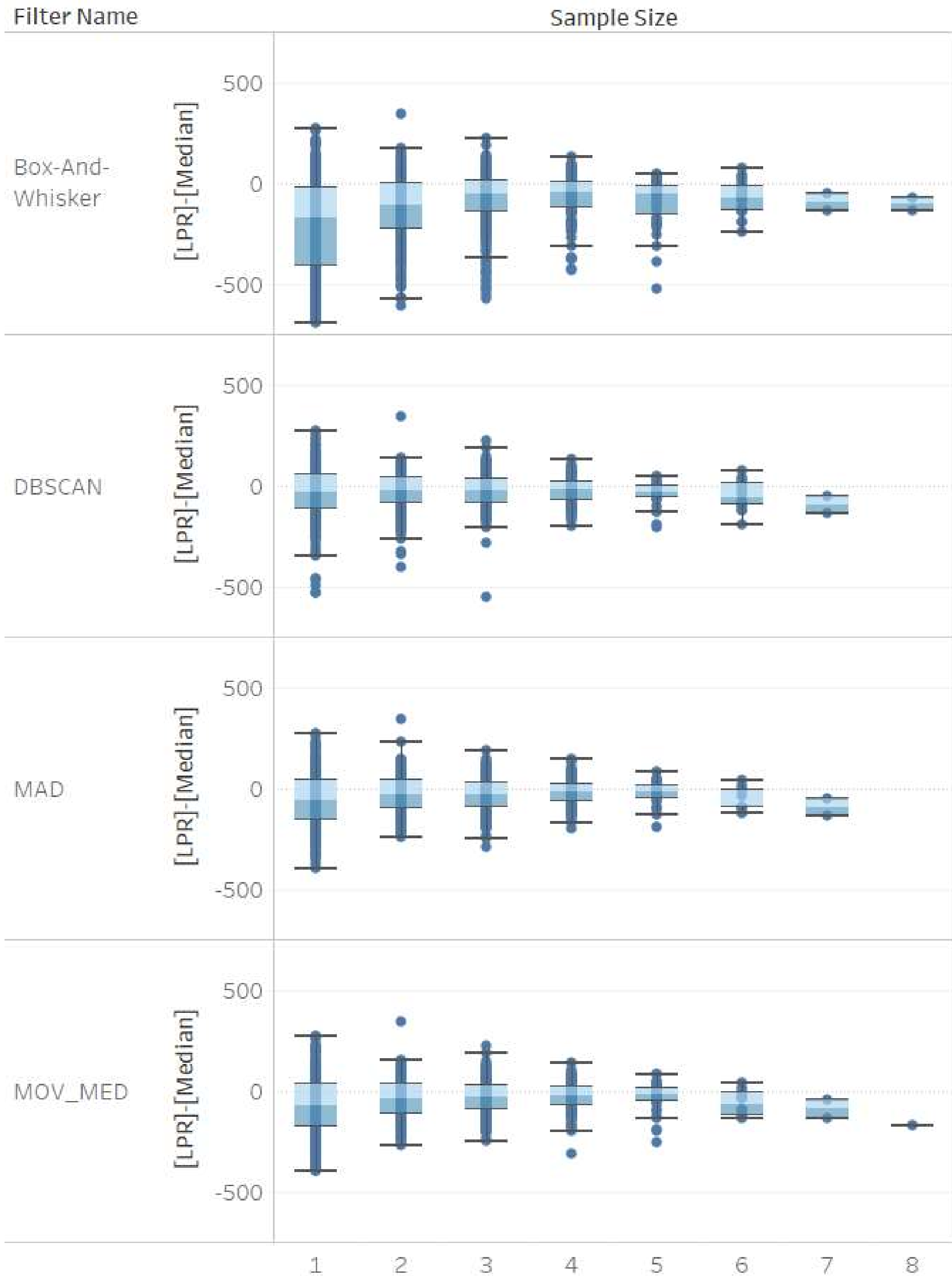

Figure 3.17 Sample Size versus Travel Time Deviation - Median

3.6 TRAVEL TIME CORRELATION

Table 3.6 Pearson correlation coefficient of travel time series data

Method / Date	Box-and-whisker	Moving Median	MAD	DBSCAN
20150401 Wed	0.403936	0.570153	0.564312	0.647593
20150402 Thu	0.195219	0.580937	0.603748	0.469232
20150403 Fri	0.369465	0.418717	0.431283	0.412606
20150404 Sat	0.088765	0.200219	0.23733	0.355395
20150405 Sun	0.395909	0.384729	0.40684	0.45904
20150406 Mon	0.312088	0.372955	0.379818	0.405874
20150407 Tue	0.045361	0.174285	0.229679	0.391101
20150408 Wed	0.193463	0.372439	0.399059	0.567928
20150409 Thu	0.16878	0.464838	0.531973	0.604589
20150410 Fri	0.162161	0.476336	0.515385	0.622431
20150411 Sat	0.552512	0.680535	0.648173	0.742513
20150412 Sun	0.153698	0.163418	0.292887	0.274784
20150413 Mon	0.190955	0.220221	0.320348	0.334782
20150414 Tue	0.1963	0.536706	0.513213	0.613606
20150415 Wed	0.203868	0.444671	0.423822	0.51505
20150416 Thu	0.144537	0.440794	0.473085	0.568709
20150417 Fri	0.169254	0.190618	0.296474	0.445956
20150418 Sat	0.035155	0.122774	0.204484	0.453778
20150419 Sun	0.185744	0.182689	0.242557	0.275823
20150420 Mon	0.101777	0.313741	0.381636	0.360786
20150421 Tue	-0.12406	0.285384	0.355358	0.425853
20150422 Wed	0.31474	0.457922	0.356702	0.383108
20150423 Thu	0.288287	0.55112	0.586613	0.680702
20150424 Fri	0.188131	0.304253	0.330413	0.466603
20150425 Sat	0.362996	0.485564	0.525865	0.486846
20150426 Sun	0.087537	0.137683	0.12062	0.287928
20150427 Mon	0.048024	0.156213	0.157298	0.33694
20150428 Tue	0.372042	0.374649	0.376825	0.50296
Average	**0.20738**	**0.359449**	**0.389493**	**0.46759**

Table 3.6 shows the Pearson correlation coefficient of a series travel time on each day calculated by MAC-address based travel time and LPR travel time. The Pearson correlation coefficient ranges from -1 to 1, a positive number means these two series datasets are positively correlated, while a negative number means negatively correlated. As shown in Table 3.6, the coefficients vary from

day to day, most of them are positive, which indicate that the MAC address-based travel time is positively correlated to the LPR travel time. Among the four filtering algorithms, DBSCAN has the highest value for most of the days, as the coefficients range from 0.27 to 0.75 with the average correlation coefficient at 0.46.

3.7 SUMMARY

In this chapter, essential steps of travel time measurement using MAC address-based travel time calculations are presented. Two of the major challenges in mobile device sensing for travel time measurement purpose, which are MAC address multiple detection matching and travel time outlier filtering, are analyzed. The advantages and application scenarios for RSSI value-based and time order-based MAC matching strategies are discussed. A new DBSCAN clustering algorithm is introduced and compared with other filtering methodologies including Moving Median Filtering, Moving Median Absolute Deviation filtering, Box-and-whisker filtering. In the experiment, different outlier filtering algorithms are applied to Acyclica dataset to measure the travel time on a 0.5-mile urban arterial in Seattle, WA. The travel time result is compared with the ground truth travel time calculated by the license plate reader data. The travel time accuracy is compared based on different variables at different scales.

To sum up, MAC address based mobile device sensing which utilizes the wireless frame from mobile devices can measure the traffic on the road. Similar traffic patterns are observed from travel time calculated by raw MAC address and LPR. However, the noise due to different travel modes and stopovers needs to be reduced by proper pairing and filtering algorithms. Order-based MAC address matching strategies are more applicable to the urban arterials with signalized intersections, and DBSCAN is most effective methods removing the travel time noises. The travel time calculated by MAC address has been proven positively correlated to the travel time calculated by

LPR. The sensor configuration, location and the valid number of samples have effects on the overall accuracy of travel time measurement.

Chapter 4. MOBILE DEVICE SENSING DATA COLLECTION FRAMEWORK

4.1 OVERVIEW

In order to collect and process the mobile device sensing data more effectively, a mobile device data collection and processing framework is introduced in this Chapter, including the client which is regarded as mobile sensors, and the server who receive, process, persist the data, calculate the travel time based on the methodology introduced in Chapter 3. This framework can automatically process the mobile device sensing data for transportation data collection in different scenarios.

4.2 OVERALL ARCHITECTURE

The mobile sensing data collection and processing framework has been designed as Client-Server architecture, as shown in Figure 4.1. At the client side, the mobile sensor can capture the data from the mobile device, preprocess and format the data. If the network connection is established and active, the client will send the data packet to the corresponding server. Otherwise, it will store the data into the internal storage. At the server side, the centralized data persistence server can handle the request, receive the data packet, verify the data integrity and persistence the data into the database. Multiple clients can establish connections to one server at the same time. The application server provides web service to end users with the integration of third-party application, in order to visualize the data and the result.

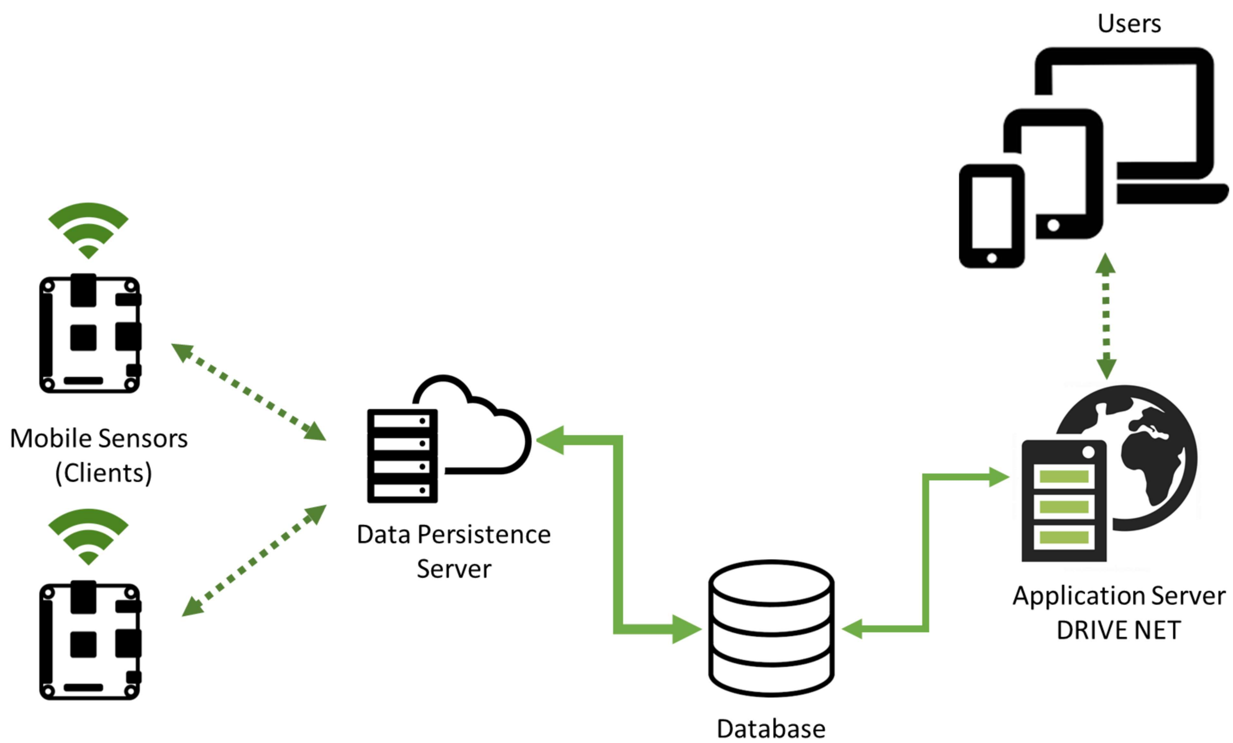

Figure 4.1 The Overall Architecture of the system

4.3 CLIENT-SIDE DESIGN

The client side of the mobile sensing system is designed to detect the signal of mobile devices in the surrounding area. Based on the type of the installed sensors and adapters, each client can detect one or two types of signals, Bluetooth and Wi-Fi. The client is also responsible for formatting and encrypting the data, establishing the connection to the server and transferring the data to the server.

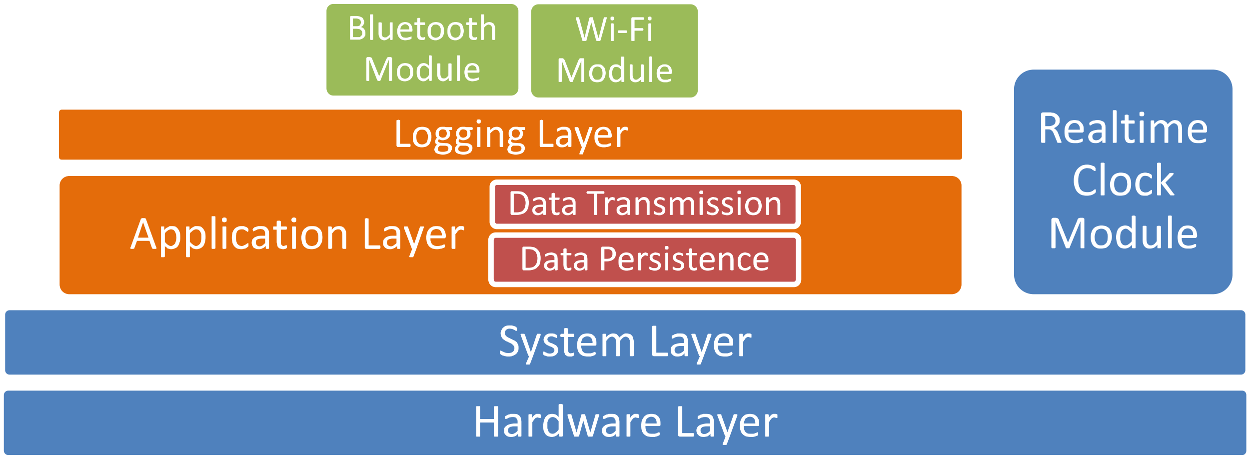

Figure 4.2 Architecture of Client Side

Figure 4.2 shows the multi-layer architecture on the client side. It contains four layers including hardware layer, system layer, application layer, logging layer and four different modules. The hardware layer of the client is based on Raspberry Pi shown in Figure 4.3. Raspberry Pi is a credit-card size computer with various type of connectors such as USB and General Purpose I/O(GPIO)[58], provides a basic processing power and potential extend ability. The system layer uses Raspbian Linux system that could be installed on the Raspberry Pi, as a result, various existing runtime library and 3rd party extensions can be applied directly. Thus, it is easy to deploy customize functions and programs on the application layer over the basic system layer.

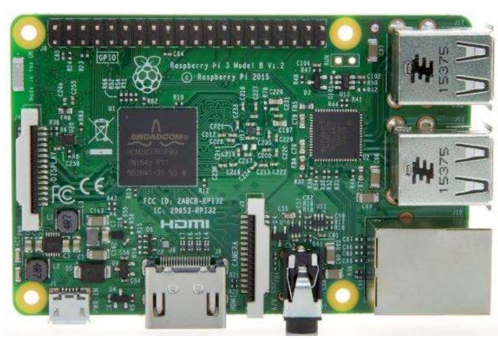

Figure 4.3 Raspberry Pi

However, there are some limitations of the Raspberry Pi module. Since the Raspberry Pi doesn't have a real time clock module onboard, whenever the device is out of power, the system time will be lost. Thus, time synchronization is required while the Raspberry Pi is starting up, which asks for reliable internet connections as the sensor startups. If the internet connection is not ensured and reliable in the working scenario, it may lead to generate data with the wrong timestamp, which is quite difficult to fix the data with correct time in the future steps. To enhance the data quality by ensuring timestamp generating correctly even the sensor is working offline, external clock module needs to be installed. The DS3231 module[59], which is an accurate I^2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator, has been plugged in the Raspberry Pi GPIO slot as a timekeeper. When the sensor is turned off, the clock module has a separate battery that can keep updating time.

The application layer of the client consists of data persistence module and data transmission module. Data persistence module aims at formatting data that receives from the hardware adapters and writes data to the internal storage. The raw MAC address would be hashed, and some user sensitive information would be removed to prevent privacy issues. Fixed prefix and suffix have been added to each data entry to help the server to check the data integrity.

Data transmission module aims at sending data from the client to the server. When the client connects to the internet, a connection request to the backend server is sent by the data transmission module. There are two different protocols could be used in the system based on different scenarios. User Datagram Protocol (UDP), which provides a procedure to send messages through the network with a minimum of protocol mechanism that can be used for broadcast connections [60], is the first strategy. The major advantage of the UDP is that the latency is low, which ensures real-time communication. However, UDP is not a 100% reliable transmission protocol, data packets might be not delivered or delivered multiple times, and there is no way to guarantee it. Thus, UDP is widely used for live audio or video transmission, which requires low latency and tolerates small amounts of data loss. As UDP is chosen as the transmission protocol, the network connection should be good and reliable, and advanced data integrity check should be applied at the server side.

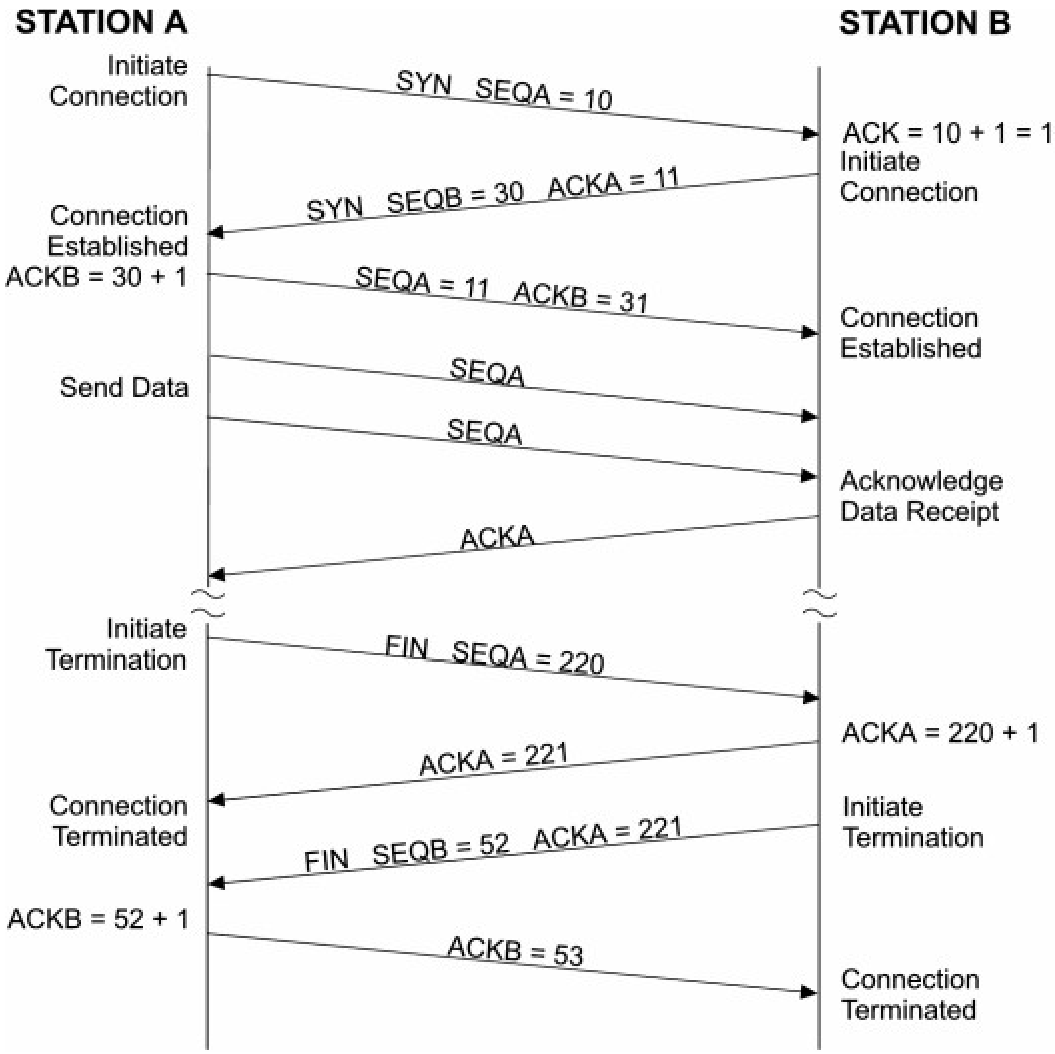

Figure 4.4 TCP data transfer process [61]

The other data transmission protocol is the Transmission Control Protocol (TCP), which ensures the delivery and the receiving order of the data packet, has been widely used to exchange files or webpages. Once TCP connection has been established between the backend server and each sensor with 3-way handshakes as shown in Figure 4.4, every data packet sent by the sender will have a sequence number SEQ. The receiver can use the SEQ to determine the order of the received data packet. Whenever the receiver gets the data, it will send an ACK package back to the sender to notify the data have been received. The advantage of TCP is the data exchange process is reliable. However, the latency might be higher and extra data packets need to be sent for verification. To avoid the whole application being stopped by the data transmission latency, the data transmission needs to execute on a different thread or system process. By using multithreading, the sensing and data transmission could be executed concurrently.

Above the application layer, a logging layer is designed to cache the data and track the status of each hardware. Wi-Fi and Bluetooth sensing module are integrated in the Raspberry Pi over the logging layer, who can detect the mobile devices such as mobile phone, routers, and onboard devices when scanning the Wi-Fi and Bluetooth signal, provide the essential requirement to collect the mobile data.

4.4 SERVE-SIDE DESIGN

The server side of the mobile sensing system is designed to receive data, verify data, insert data into the database and utilize data for real-time analysis purpose. It has three major components, data persistence server, database and web application server.

4.4.1 *Data Persistence Server*

The data persistence server is designed for receiving and validating data from the client side. It listens on a specific port for the connection of the client. Once it receives the connection request from the client, the server will create a new thread to receiving the data. Hence, multiple clients can connect to the server simultaneously.

Figure 4.5 The Process on Data Persistence Server

Figure 4.5 shows the overall process on data persistence server. When the data is received from the client-side, it will be verified and extracted by the Regular Expressions, a method for searching and locating specific substring in a large portion of text [62]. Timestamp, Sensor ID, Data Series, Hashed MAC address, RSSI value and detection mode will be extracted from the data packet. A connection between thread and database server will be established. Considered the database has a maximum connection limit, the thread will acquire and release the database connection handle through the connection pool. If all the connection handles are occupied, the thread will store the data into the temporary buffer first to prevent data loss. Once the connection handle is available, the extracted data will be inserted into the database.

4.4.2 *Database*

The database is designed as a middleware to persistent and provide the data. It receives the data from the persistence server and sends the result based on the query to the application server. The read operation and write operation have been separated. Postgres Server, one of the relational database systems[63], have been selected. It can be extended with PostGIS, which is one of the popular Geospatial databases, to support to process geographical data[64]. Database has been divided into 4 different tables, as shown in Figure 4.6 using Unified Modeling Language(UML) diagram[65], and the detailed schema of these tables is showed in Table 4.1-Table 4.4.

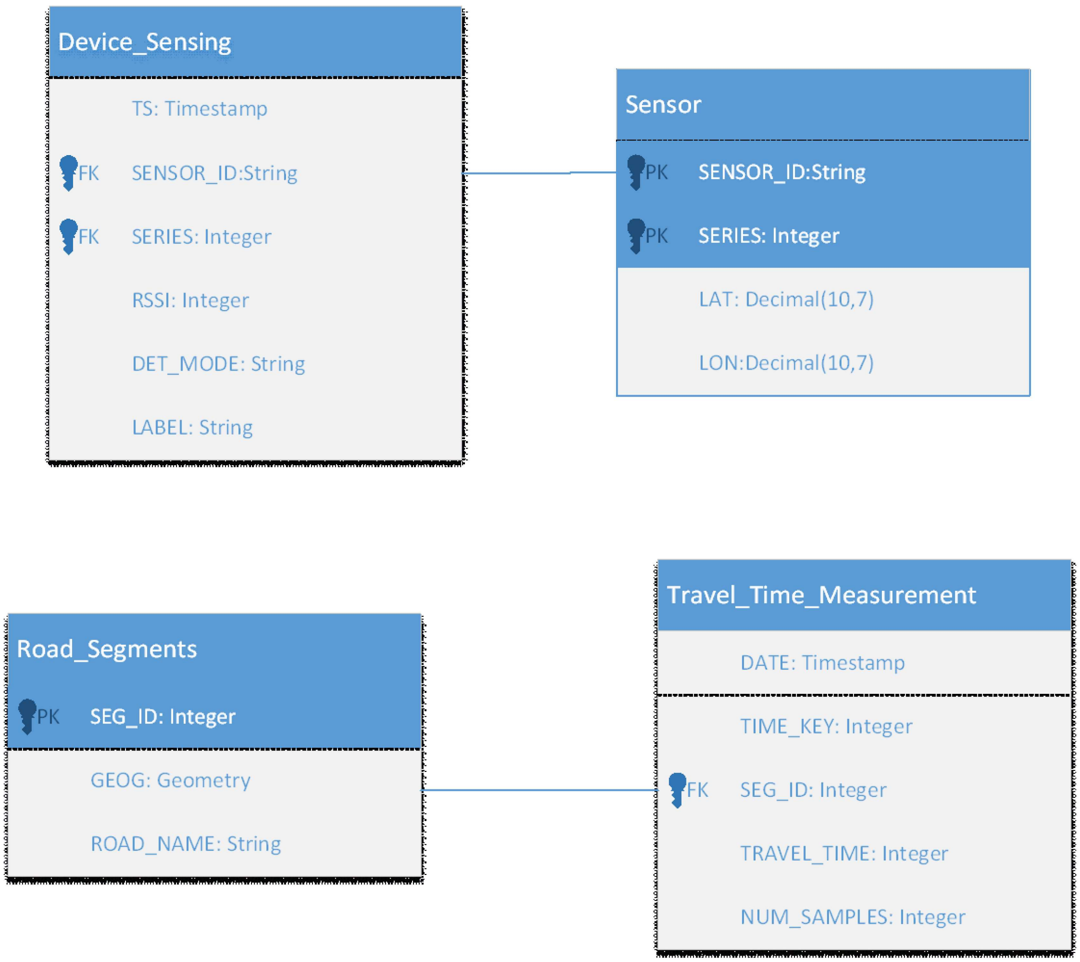

Figure 4.6 Database design UML diagram

Table 4.1 Database Schema for device sensing data

Data Field	Data Type	Description
TS	Timestamp	the time point as the data were collected.
SENSOR_ID	String	Refers to sensor table, indicates which sensor collected the data
SERIES	Int	Data Series, represents the location/purpose of data collection
RSSI	Int	RSSI signal intensity received by the sensor
DET_MODE	String	Detection Mode: Bluetooth/wifi
LABEL	String	Status of the data entry: unprocessed, accepted or filtered.

Table 4.2 Database Schema for Sensor

Data Field	Data Type	Description
SENSOR_ID	Int	ID of Sensor
SERIES	Int	Data Series
LAT	Decimal (10,7)	Latitude
LON	Decimal (10,7)	Longitude

Table 4.3 Database Schema for Travel Time Measurement

Data Field	Data Type	Description
DATE	Timestamp	Date
TIME_KEY	Int	The order of 5-minutes period in one day, ranges from 0 to 287
SEG_ID	Int	Road segment ID, refers to the road table
TRAVEL_TIME	Int	Travel time in seconds
NUM_SAMPLES	String	Number of valid samples

Table 4.4 Database Schema for Road Segments

Data Field	Data Type	Description
SEG_ID	Int	Road segment ID
GEOG	Geometry	Road Geometry
ROAD_NAME	String	Name of Road Segment

4.4.3 *Application Server*

Application server provides a user-friendly GUI web interface to end users. The travel time data and road scenarios can be visualized and combine with other data sources or 3rd party interactive map services such as Leaflet Map, Openstreet Map on the application server. With the capability of application server, users can directly query the road segment, have access to the travel time data. Figure 4.7 shows an example of GUI design to visualize the travel time variation using line charts.

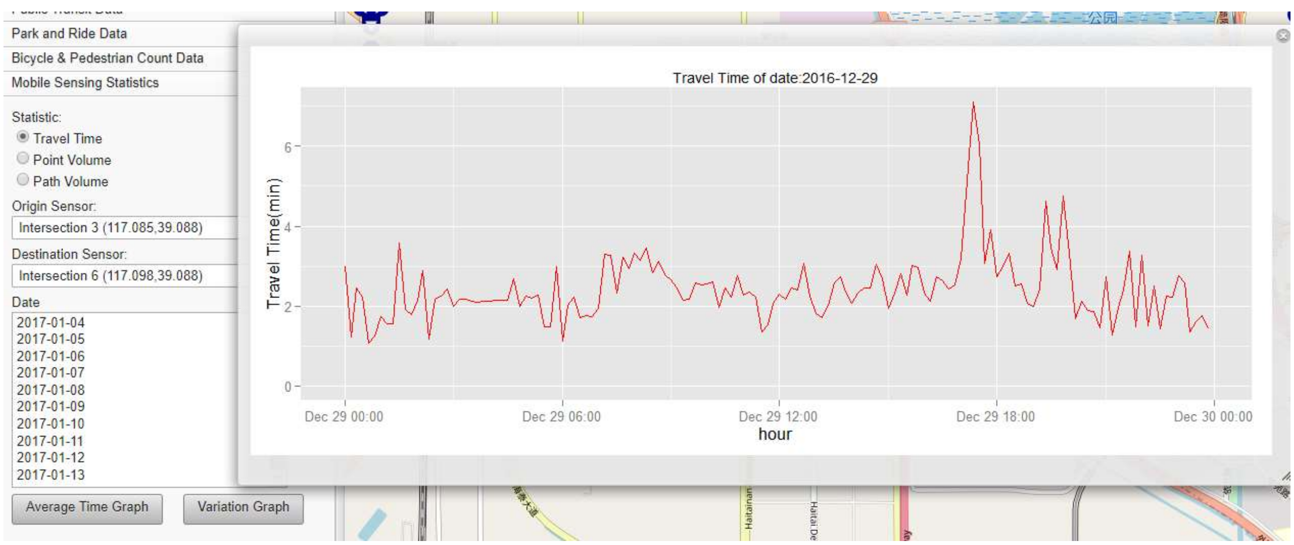

Figure 4.7 Using Line Chart to Visualize Travel Time Variation

4.5 SUMMARY

In this chapter, a mobile sensing data collection and processing framework is proposed as Client-Server architecture. At client side, a multi-layer architecture is presented to be deployed on the Raspberry Pi devices that can capture the mobile device data, ensure the data integrity, monitor the status of the sensor and transport the data to the backend server. At the server side, three major components, as data persistence server, database, and application server are introduced to process, persistence, and visualize the MAC address and travel time result. This framework can be used for most of the scenarios to process mobile device sensing data for travel time measurement.

.

Chapter 5. IMPACT OF MAC ADDRESS RANDOMIZATION

5.1 OVERVIEW

Mobile device sensing technology, which uses MAC address as a device identifier to recognize devices, has been addressed privacy concerns [66]. Although all the data is collected before the connection established, the public still worried about being tracked by their own devices. Some cellular device manufacturers have raised an idea that using random MAC address instead of hardware MAC address to reduce the potential risk. In 2014, Apple, Inc. launched iOS 8.0 operating system with new function MAC address randomization [67]. The Wi-Fi scanning behavior of the cellular devices has changed to use random, locally administrated MAC addresses instead of hardware MAC addresses in the probe request and response frame [4]. Later, Android introduced an interface for MAC randomization in Android 6.0 [68], although it was not fully implemented by all manufacturers. Windows 10 also has a random hardware address configuration, which enables devices to use random MAC addresses for both probe requests and Wi-Fi connections [69]. The MAC randomization algorithms and strategies vary a lot. With these new updates, a challenging problem which arises in the transportation mobile sensing data is that the devices with MAC randomization cannot be tracked by current sensors. There is a further problem that MAC randomization strategy may have a great impact on the data quality and the result of the MAC address based mobile sensing data.

Currently, only a few studies have shown the impact of the MAC randomization on the mobile sensing data collection. However, the influence on the data quality especially in the transportation area has rarely been examined directly. In order to evaluate the impact of MAC randomization on mobile device sensing, different datasets have been applied to measure the proportion of the random MAC address.

In the past research, Matte raised a method that using Information Elements (IEs) instead of MAC address to fingerprinting devices[23]. An idea that using IEs as the identifier of mobile devices may have the ability to mitigate the impact of MAC address randomization in transportation data collection has been raised. To verify the effectiveness of this idea, an experiment is conducted on Burke Gilman Trail in Seattle to capture the MAC addresses and IEs from pedestrian and cyclists. The experiment design and evaluation process are explained in Section 5.6.

5.2 MAC RANDOMIZATION MECHANISM

The MAC address randomization mechanism varies a lot. It requires support by the network card driver and the operating system[23], highly depends on the device model and vendors. According to previous researches, some devices change their MAC addresses every probe request, while others may keep their MAC address for several iterations.

5.2.1 *Android*

Android introduces a MAC randomization interface in Android 6.0. When a device initiates a background Wi-Fi or Bluetooth scan, the operation is visible to external devices as originating from a randomized MAC address [68]. However, it requires the support from network card driver. Martin et al noticed that Samsung devices didn’t perform MAC address randomization, possibly due to the chipset combability issues [21]. Most of randomized MAC address from Android phone will share the same prefix DA:A1:19, which is a CID owned by Google. Others may keep their origin CID or use the manufacture’s CID with the local bit set. The CID for randomized MAC address is also configurable in the system [21]. In a word, Android devices only perform 24-bit randomization, and most Android devices using default settings may share the same CID.

Matte did a case study on MAC address behavior for Nexus 6P, Nexus 5S and OnePlus 3000, found that the random addresses for these Android phone changes every probe requests, but lots of addresses have been reused more than once [23]. MAC addresses change so frequently that they can prevent devices being tracked and recognized correctly by mobile sensing sensors. The reuse pattern for MAC addresses may have a higher collision rate between devices who share the same chipset for Android Phone, which may mismatch different devices to the same MAC address when collecting the data.

5.2.2 *iOS*

MAC Randomization for iOS devices first implemented in iOS 8. It was only used when the phone waked up from a sleep mode without a Wi-Fi connection [70]. However, Vanhoef et. found that the iOS 10 and later versions always use random addresses for probe requests even devices were active [22]. Unlike Android devices, who share the same CID for the first 24-bit of MAC address, iOS devices use random CIDs with only the local and broadcast bit set, which indicate that iOS performs 46-bit randomization. Since iOS is not an open source operating system, there are no researches revealed the randomization mechanism of iOS devices, neither obvious pattern has been observed. The random MAC addresses seems to be distributed uniformly [21]. Matte have found that most of addresses are used for more than one probe requests, the same as what has been observed in our experiment.

5.3 STUDY DATA

To understand the proportion of randomized MAC addresses and investigate the impact, a historical MAC address dataset, collected by mobile device sensing system for traffic monitoring in Tianjin, China, is selected to serve as the primary data source. This system which started in

2016 is used to monitor the real-time traffic status in Tianjin Binhai New Area. It consists of 16 different sensor units at different intersections that can detect both Bluetooth and Wi-Fi signals. The sensor locations are shown in Figure 5.1. We sampled 6 days of data in the 6-month period, from November 2016 to January 2018, to measure the trend of MAC randomization.

Figure 5.1 Mobile Sensing Traffic Measurement System in Tianjin, China

Table 5.1 Data Sample of Tianjin Mobile Sensing Dataset

SensorID	timestamp	MAC	Type	SSID
15	11/18/2016 3:53	18:DC:56:B6:73:62	Bluetooth	0
14	11/18/2016 3:53	48:02:2A:D4:B7:92	Wifi	-65
14	11/18/2016 3:45	B8:27:EB:97:0C:5C	Wifi	-19
14	11/18/2016 3:45	48:02:2A:D4:B7:92	Wifi	-65
15	11/18/2016 3:53	18:DC:56:B6:73:62	Bluetooth	0
15	11/18/2016 3:53	18:DC:56:B6:73:62	Bluetooth	0
15	11/18/2016 3:53	18:FE:34:A5:E2:B6	Wifi	-85
14	11/18/2016 3:45	48:02:2A:D4:B7:92	Wifi	-65
15	11/18/2016 3:53	18:DC:56:B6:73:62	Bluetooth	0
15	11/18/2016 3:53	18:DC:56:B6:73:62	Bluetooth	0

The data sample of the Tianjin Mobile Sensing dataset is shown in Table 5.1
Table 5.1. Tianjin mobile sensing dataset contains 5 different fields: SensorID, timestamp, MAC, type and SSID. The dataset collects both Bluetooth and Wi-Fi sensing data, and each SensorID is associated with one specific sensor at one location. Based on the timestamp and MAC address, the travel time can be calculated. The advantage of this dataset is that consists of raw MAC addresses, which can be used to recognize whether the address is a locally assigned MAC address.

5.4 METHODOLOGY

5.4.1 *Global MAC Address Separation*

It is of importance to recognize whether a collected MAC address is a random address or not. However, it is hard to identify the random MAC address directly since different mobile phone OS implemented randomization in different ways. One of the methods developed by Martin et al [21] to identify random MAC addresses requires additional information such as Wi-Fi Protected Setup values and the 802.11 Information Elements (IEs), which is not applicable in our dataset. However, the universal MAC address should be unique among all the devices subjected to IEEE 802.11 standard. Each device connected to the network should have a unique MAC address. The MAC address format is shown in Figure 5.2.

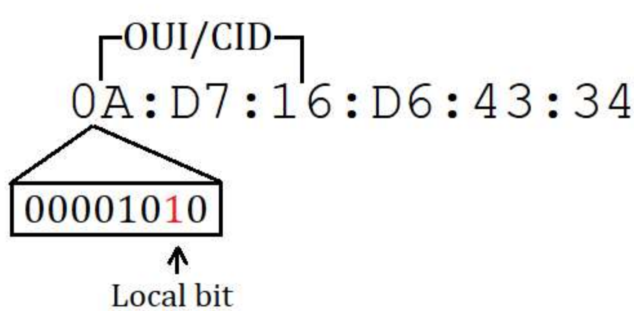

Figure 5.2 MAC Address Format

IEEE Registration Authority (IEEE RA) assigned the higher 24-bit as the (OUI) or the Company ID (CID), which represents the vendor, manufacturer or organization of the network hardware [71]. The registered OUI or CID can be found in the IEEE database. For each assigned OUI or CID, the lower 24-bit of MAC address is a serial number, thus 224 distinct spaces are given to the specific vendor to register their network devices. The MAC address is usually presented in six octets with the separation ":" or "-", such as "0A:D7:16:D6:43:34".

In addition to the global unique MAC address, the device can use locally assigned MAC address with the Local bit on which are not ensuring to be unique, which represent the device uses their universal (X=0) or local address (X=1). It implies that the random generated MAC address should be locally assigned in most cases. Based on the previous research by Martin et al [21], the local bit could be used to identify whether the device uses their hardware MAC address or random MAC address. Thus, in this paper, we categorize the MAC address as either local or universal address by checking the Local bit in the CID.

5.4.2 *Trip Sampling Rate*

To measure the travel time based on the mobile device data, the same MAC address must be observed and paired over different places. Then the travel time for this MAC addresses could be calculated based on the difference of the timestamp captured by these two sensors. However, for the devices with MAC randomization capability, the MAC address may not be paired if the device changes the MAC address before it arrives the second sensor. In a word, MAC address is not effective for travel time estimation unless it can be paired over multiple sensors in the system. Daily Trip sampling rate (TSR) is a designed criterion to measure the effectiveness of the MAC data collection between a pair of mobile sensors on the same day. TSR is defined in Equation (11).

$$\text{TSR} = \frac{\textit{Number of Trips}}{\textit{Number of Distinct MAC addresses}} \times 100\% \tag{11}$$

After analyzing the data in Tianjin Mobile Sensing Traffic Measurement System from Jun 2017 to Dec 2017, the overall average TSR is around 7%. Except for MAC address randomization, TSR is also affected by the sensor configuration such as Antenna, installed location, detection rate [13], the traffic pattern as speed and volume, and the penetration rate of the mobile devices. To mitigate the uncertainty of the sensor configurations, two sensors on a specific road segment are selected to do more experiment. The total number of distinct collected MAC address is counted, and the number of trips paired by these two sensors is calculated on each day. MAC addresses and trips are classified into 4 exclusive bins depending on the address type (Universal/Local) and detection type (Wi-Fi/Bluetooth). Linear regression models are built for each bin to measure the TSR with different conditions.

5.4.3 *Collision Rate*

MAC randomization also increases the possibility of mismatching two different devices as the same one if these two devices have the same randomized MAC address in a short period. Because the randomization mechanism varies a lot and the constraint of data availability, we would evaluate the ideal collision rate based on the number of randomized bits to simplify the problem. If all the randomized MAC addresses are uniformly distributed, the large number of randomized bits, the less probability to have a collision.

As the problem is exactly the same as the birthday problem [72] in probability theory, if there are N random MAC addresses with K-randomized bits, the equation for calculating collision rate is shown in Equation(12).

$$P(N,K) = 1 - \frac{(2^K)!}{(2^K)^N(2^K-1)} \quad (12)$$

P(N,K) represents the probability of MAC address collision.

5.5 Result

5.5.1 *Global MAC Address Separation*

Figure 5.3 shows the proportion of local and universal MAC addresses over time. It is obvious that the proportion of the local MAC address is growing from less than 20% in November 2016 to about 50% in July 2017, and then keeping consistent around 50% until January 2018. The unusual pattern of the increasing proportion of local MAC address is likely due to wider usage of MAC randomization. Since some of the mobile device vendors still not support MAC randomization, if more vendors apply this technology, the proportion of the local MAC address may still increase.

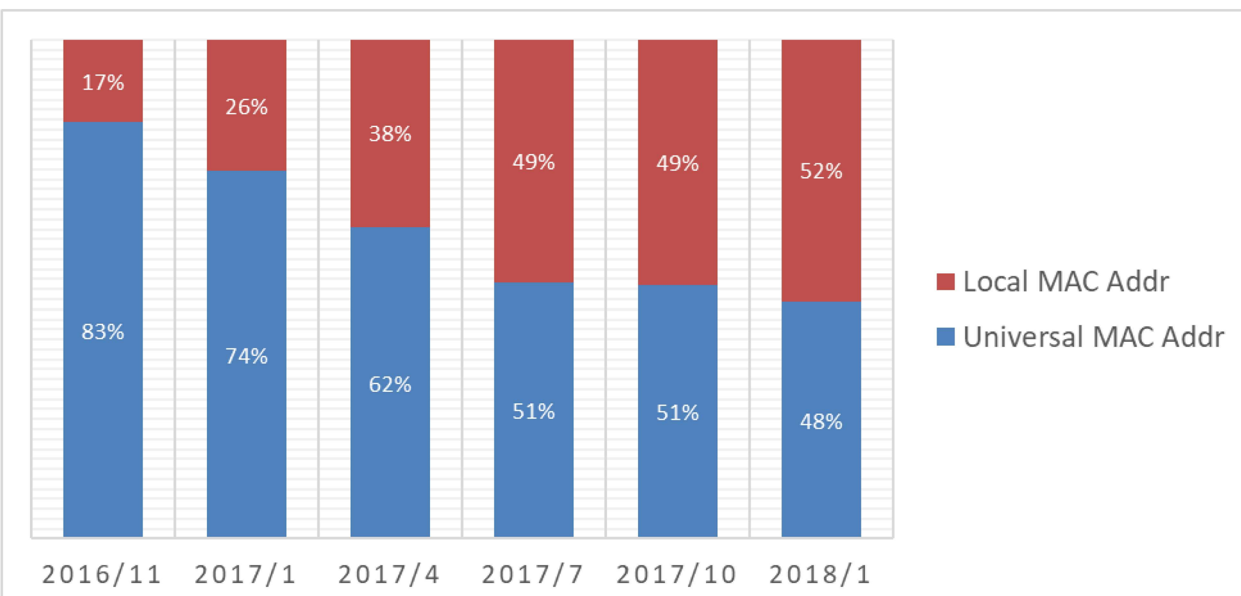

Figure 5.3 Universal/Local MAC Address Collected by Roadside Sensor

To ensure the source of the local MAC addresses is mobile devices rather than other routers or P2P service devices, the CID of the MAC address is used to find the manufacture of the network devices. The result is compared with the smartphone market share to the device manufacture derived by the MAC address. Refers to the IDC Quarterly Mobile Phone Tracker 2017 Q3 Report [28], the market share of the top five smartphone manufactures is showed in Figure 5.4(A). And the manufacturer based on the CID of the MAC address is showed in Figure 5.4(B). Except Oppo, the proportions of other four major manufactures recognized by MAC address are significantly lower than the expected market share, which indicates that the random generated MAC address didn't use the same CID as their hardware MAC address. The previous study discovered that most of the randomization MAC address of Android Phone would share the DA:A1:19 prefix [21], which is an OUI owned by Google. This prefix can be a good indicator to identify the potential random MAC address.

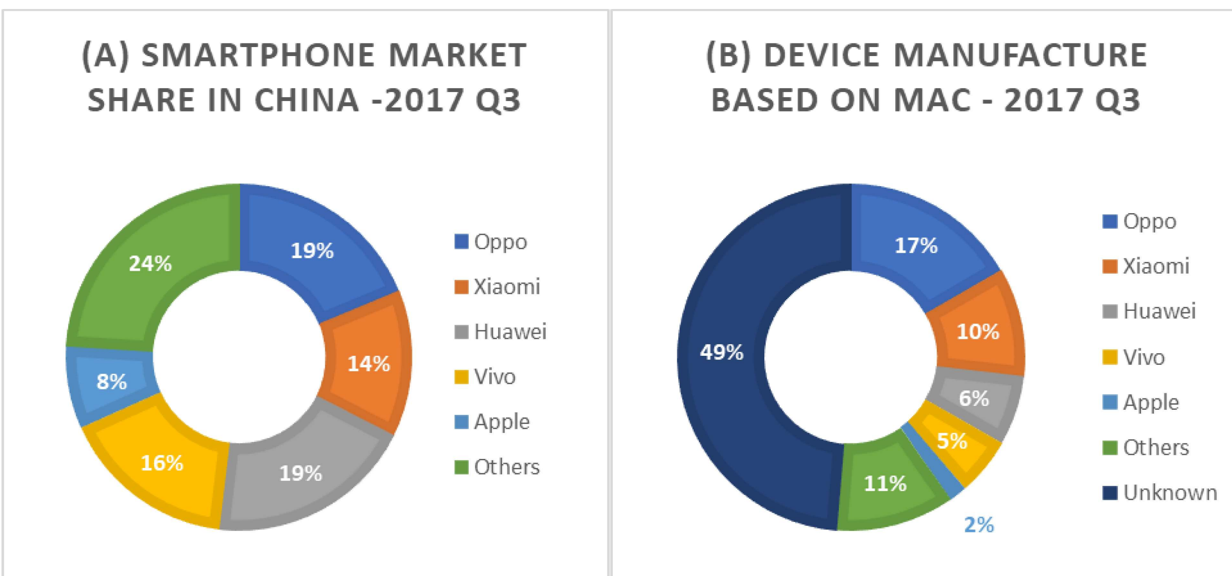

Figure 5.4 Manufacture of the Mobile Device

5.5.2 *Trip Sampling Rate*

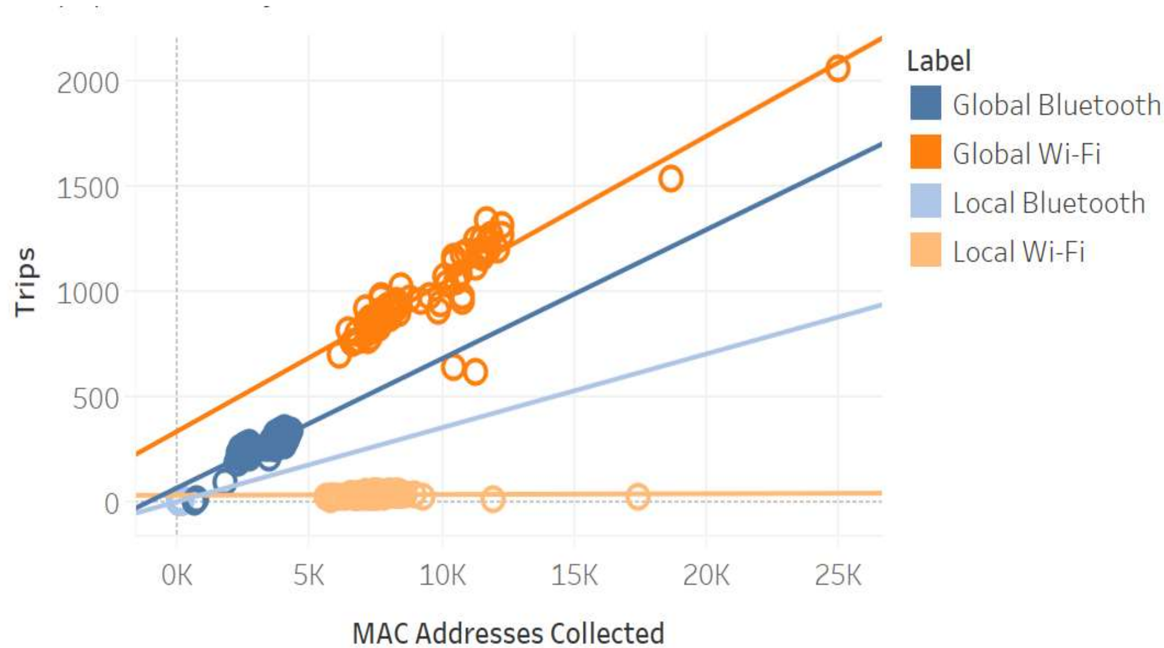

Figure 5.5 TSR Linear Regression Model

As shown in Table 5.2 and Figure 5.5, all the TSR coefficients are statistically significant, have proven the linear relation between the number of collected MAC addresses and trips. The TSR for universal MAC address that detected by Wi-Fi is 7.98%, while is only 0.5% for local MAC addresses. For Bluetooth, the TSR of universal MAC address is also higher than the local MAC address. To sum up, the local MAC addresses have less contribution to the paired trips which are essential for travel time estimation. As the proportion of the local MAC addresses was getting larger, with the same amount of collected data, the number of the sampling trips would be decreased.

Table 5.2 Trip Sampling Rate

	Universal MAC				Local MAC			
		Estimate	Std. Error	P-value		Estimate	Std. Error	P-value
Wi-Fi	(Intercept)	255	4.43E+01	1.95E-07	(Intercept)	-5.6796	6.98087	0.419
	TSR	0.0798	4.79E-03	< 2e-16	TSR	0.00559	0.00092	6.12E-08
Bluetooth	(Intercept)	135	1.26E+01	<2e-16	(Intercept)	2.41338	1.10362	0.032008
	TSR	0.04205	3.72E-03	<2e-16	TSR	0.03637	0.00931	0.000209

5.5.3 *Collision Rate*

It is obvious that the larger the number of randomized bits, the lower the collision rate. As we found that Android devices share the common prefix, mostly DA:A1:19, have a higher collision rate than the iOS devices, which perform 46-bit randomization. The collision probability for Android devices with the same prefix is shown in blue in Figure 5.6. If there are more than 4800 devices in a mobile sensing system, there are around 50% probability that two devices perform the same randomized MAC address.

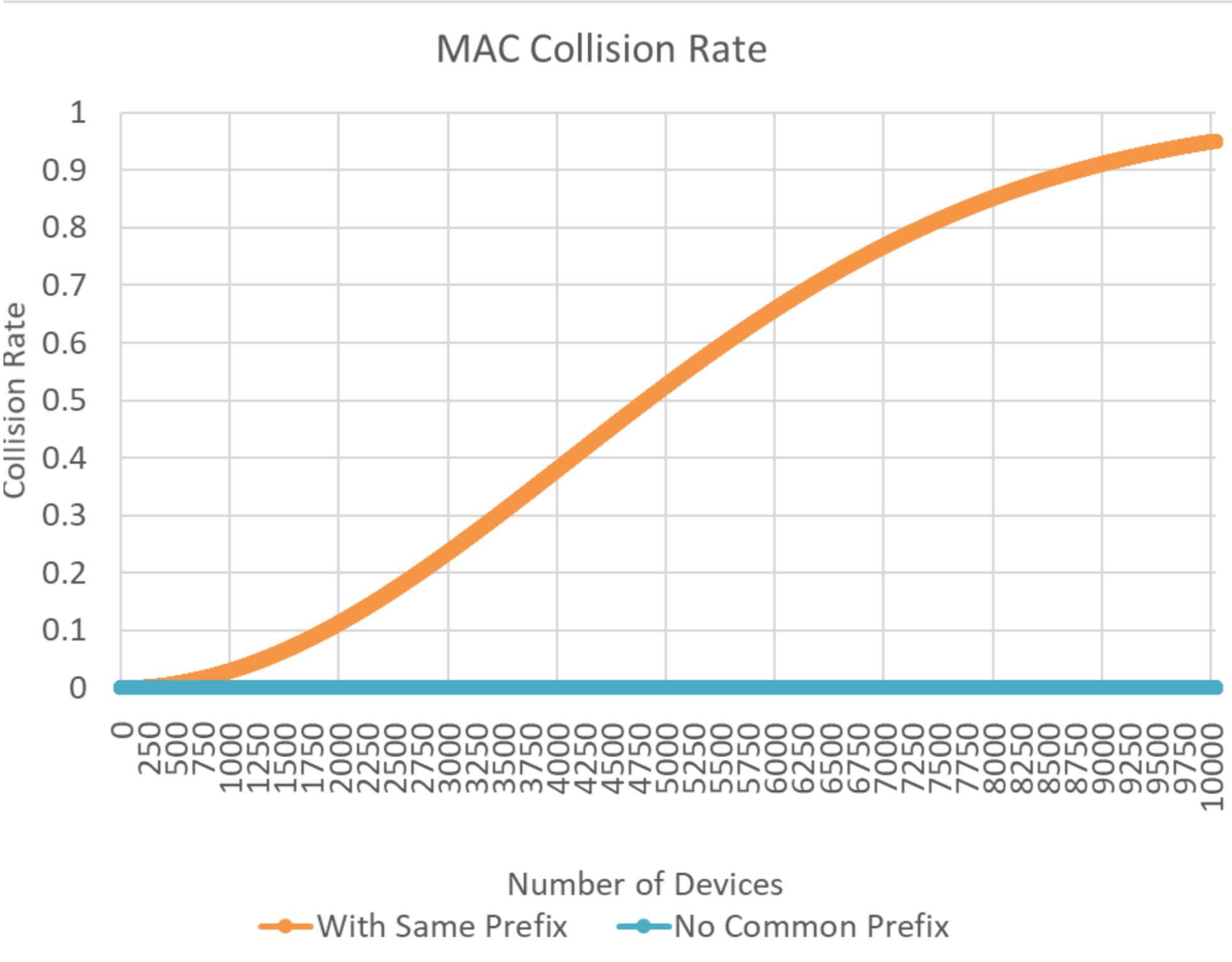

Figure 5.6 Collision Rate for Android Devices

5.6 DEVICE RECOGNITION USING INFORMATION ELEMENTS

5.6.1 *Overview*

Information elements (IEs), also called tagged parameters, are variable-length fields which contained in the probe request frames regulated by IEEE 802.11 specification[58]. Figure 5.7 shows the format of the probe request frame, which consists of two major parts, MAC header, and the frame body. The MAC header contains the identification of the devices that remains the same for all the frames, while the frame body is different based on the specific frame type. The frame body contains the SSID, supported rates and other optional tagged parameters which represent the extended functionality of the devices. The IEs are not mandatory and various from devices to devices depending on the configuration and capabilities of the device, which may have the potential ability to identify and fingerprint devices[23].

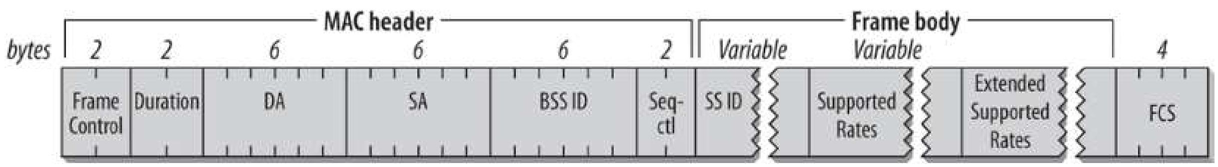

Figure 5.7 Probe Request Frame[58]

5.6.2 *Information Elements Extracting*

To extract the information elements, there are several packet sniffers and network analyzers available, such as SolarWinds, Paessler, Wireshark, etc [75]. To extract the information elements more effectively, two base modules TShark and Panoptiphone are selected for this research. TShark, as a lightweight, terminal oriented version of Wireshark designed for capturing and analyzing network packets[76], can be easily integrated into other modules. Figure 5.8 shows the raw output of tagged results in probe request frames in Extensible Markup Language (XML) format extracted by TShark.

```
      <field name="wlan.tag" showname="Tag: DS Parameter set: Current Channel: 10" size="3" pos="54" show="" value="">
        <field name="wlan.tag.number" showname="Tag Number: DS Parameter set (3)" size="1" pos="54" show="3" value="03"/>
        <field name="wlan.tag.length" showname="Tag length: 1" size="1" pos="55" show="1" value="01"/>
        <field name="wlan.ds.current_channel" showname="Current Channel: 10" size="1" pos="56" show="10" value="0a"/>
      </field>
      <field name="wlan.tag" showname="Tag: HT Capabilities (802.11n D1.10)" size="28" pos="57" show="" value="">
        <field name="wlan.tag.number" showname="Tag Number: HT Capabilities (802.11n D1.10) (45)" size="1" pos="57" show="45"
        <field name="wlan.tag.length" showname="Tag length: 26" size="1" pos="58" show="26" value="1a"/>
        <field name="wlan.ht.capabilities" showname="HT Capabilities Info: 0x016e" size="2" pos="59" show="0x0000016e" value="
          <field name="wlan.ht.capabilities.ldpccoding" showname=".... .... .... ...0 = HT LDPC coding capability: Transmitter
unmaskedvalue="6e01"/>
          <field name="wlan.ht.capabilities.width" showname=".... .... .... ..1. = HT Support channel width: Transmitter suppo
01"/>
          <field name="wlan.ht.capabilities.sm" showname=".... .... .... 11.. = HT SM Power Save: SM Power Save disabled (0x3)
          <field name="wlan.ht.capabilities.green" showname=".... .... ...0 .... = HT Green Field: Transmitter is not able to
maskedvalue="6e01"/>
          <field name="wlan.ht.capabilities.short20" showname=".... .... ..1. .... = HT Short GI for 20MHz: Supported" size="2
          <field name="wlan.ht.capabilities.short40" showname=".... .... .1.. .... = HT Short GI for 40MHz: Supported" size="2
          <field name="wlan.ht.capabilities.txstbc" showname=".... .... 0... .... = HT Tx STBC: Not supported" size="2" pos="5
          <field name="wlan.ht.capabilities.rxstbc" showname=".... ..01 .... .... = HT Rx STBC: Rx support of one spatial stre
          <field name="wlan.ht.capabilities.delayedblockack" showname=".... .0.. .... .... = HT Delayed Block ACK: Transmitter
alue="6e01"/>
          <field name="wlan.ht.capabilities.amsdu" showname=".... 0... .... .... = HT Max A-MSDU length: 3839 bytes" size="2"
          <field name="wlan.ht.capabilities.dsscck" showname="...0 .... .... .... = HT DSSS/CCK mode in 40MHz: Won&#x27;t/Can&
"6e01"/>
          <field name="wlan.ht.capabilities.psmp" showname="..0. .... .... .... = HT PSMP Support: Won&#x27;t/Can&#x27;t suppo
          <field name="wlan.ht.capabilities.40mhzintolerant" showname=".0.. .... .... .... = HT Forty MHz Intolerant: Use of 4
edvalue="6e01"/>
          <field name="wlan.ht.capabilities.lsig" showname="0... .... .... .... = HT L-SIG TXOP Protection support: Not suppor
        </field>
        <field name="wlan.ht.ampduparam" showname="A-MPDU Parameters: 0x03" size="1" pos="61" show="0x00000003" value="03">
          <field name="wlan.ht.ampduparam.maxlength" showname=".... ..11 = Maximum Rx A-MPDU Length: 0x3 (65535[Bytes])" size=
          <field name="wlan.ht.ampduparam.mpdudensity" showname="...0 00.. = MPDU Density: no restriction (0x0)" size="1" pos=
          <field name="wlan.ht.ampduparam.reserved" showname="000. .... = Reserved: 0x0" size="1" pos="61" show="0x00000000" v
        </field>
        <field name="wlan.ht.mcsset" showname="Rx Supported Modulation and Coding Scheme Set: MCS Set" size="16" pos="62" show
          <field name="wlan.ht.mcsset.rxbitmask" showname="Rx Modulation and Coding Scheme (One bit per modulation): 2 spatial
            <field name="wlan.ht.mcsset.rxbitmask.0to7" showname=".... .... .... .... .... .... 1111 1111 = Rx Bitmask Bits 0-
            <field name="wlan.ht.mcsset.rxbitmask.8to15" showname=".... .... .... .... 1111 1111 .... .... = Rx Bitmask Bits 8
            <field name="wlan.ht.mcsset.rxbitmask.16to23" showname=".... .... 0000 0000 .... .... .... .... = Rx Bitmask Bits
            <field name="wlan.ht.mcsset.rxbitmask.24to31" showname="0000 0000 .... .... .... .... .... .... = Rx Bitmask Bits
            <field name="wlan.ht.mcsset.rxbitmask.32" showname=".... .... .... .... .... .... .... ...1 = Rx Bitmask Bit 32: 0
            <field name="wlan.ht.mcsset.rxbitmask.33to38" showname=".... .... .... .... .... .... .000 000. = Rx Bitmask Bits
            <field name="wlan.ht.mcsset.rxbitmask.39to52" showname=".... .... ...0 0000 0000 0000 0... .... = Rx Bitmask Bits
/>
            <field name="wlan.ht.mcsset.rxbitmask.53to76" showname="...0 0000 0000 0000 0000 0000 000. .... = Rx Bitmask Bits
0"/>
          </field>
```


Figure 5.8 Information Elements in XML format from TShark

However, since the raw TShark result contains the information elements as well as other tagged parameters in the frame, and the hierarchical XML result is good for organizing but less efficient to store and search the data. Another module Panoptiphone is applied to post-process the result of TShark. Panoptiphone [77], which is a python module developed by Matte and Cunche, originally used for device fingerprinting, can parse the XML output from tshark to a list of information elements. By Integrating the TShark, Panoptiphone and other python modules, the information elements could be extracted from the probe request frames as the first step of the device identification.

5.6.3 *Device identification*

Device identification is aimed at using IEs as a complimentary information to enhance the MAC address-based identification. According to the observation, IEs keep the same value while the mobile device is performing MAC randomization as shown in Table 5.3.

Table 5.3 Random MAC address with same Information Elements

MAC Address	Hashed_IEs
0e:57:91:90:1c:87	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
0e:57:91:90:1c:87	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
a2:1e:89:85:e2:26	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
a2:1e:89:85:e2:26	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
76:e5:cb:52:7e:e1	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
76:e5:cb:52:7e:e1	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
ca:e8:55:f3:e2:88	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
ca:e8:55:f3:e2:88	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
ce:7f:81:1e:c0:79	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b
02:f4:e9:69:40:a5	7ef2cf8628f4d9fc330cd47a651a0f3fd02acf4e74610dbc1a39fb20698bad4b

In past studies, Vanhoef selected 12 different IEs as a combination to fingerprinting devices[22], although not all devices could be uniquely recognized by these methods, IEs are still good indicators to identify devices. In this research, the method is extended by using most of the IEs for device recognition except whom are not stable and may vary between different probe requests for the same device, to decrease the possibility of the false negative. A hashed algorithm is used to compare the IEs between different devices.

5.6.4 *Study Location*

The study location is selected as a 0.2-mile segment on Burke Gilman Trail in University of Washington, a multi-use path in North Seattle shown in Figure 5.9. Mobile sensors are installed at each end of the study segment to collect IEs and MAC addresses. Based on the Google Map, the average travel time for pedestrians is 4 minutes, while 2 minutes for cyclists. The data collection lasts for 40 minutes during the PM peak hour.

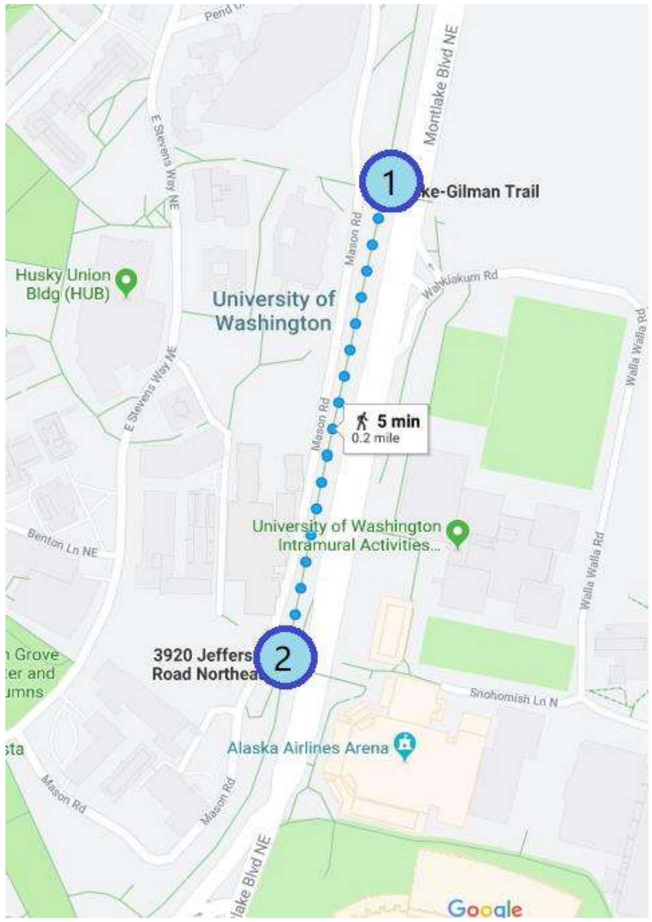

Figure 5.9 Study Location

5.6.5 *Results*

Table 5.4 shows the summary of MAC address and IEs data collection. Sensor 1 has collected 69 unique MAC addresses during the time period as sensor 2 have collected 354 addresses. The number of IEs collected is larger than the global MAC address but less than the number of local MAC address, which verifies the assumption that different MAC addresses may have the same IEs.

Table 5.4 Data Summary

	Unique # of Global MAC	Unique # of Local MAC	Unique # of IEs
Sensor 1	34	35	44
Sensor 2	122	232	141

After MAC address matching step, 23 MAC addresses have been observed by both Sensor 1 and Sensor 2, while 19 IEs are captured by both sensors. Each MAC address only associated with one group of IEs. To understand whether IEs can distinguish the unique devices, a Gantt Bar plot is used to visualize the detections. Figure 5.10 shows a valid trip associated with the same IEs and MAC address recognized by three detections. The X-axis is the timestamp, and Y-axis represents the ID of the sensor. This device left sensor 1 at 3:59:59 PM and arrived sensor 2 at 4:03:27 PM which is a trip recognized by mobile devices sensing data.

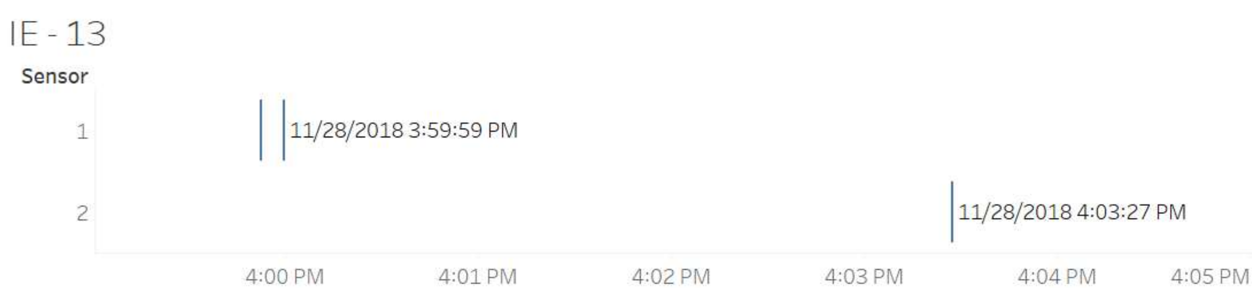

Figure 5.10 Valid Trip Recognition

However, most of the information elements have been observed belonging to different mobile devices, as shown in Figure 5.11. The frequency and duration of the detection are too high. Moreover, both sensors have observed the same IEs value at the same time.

Figure 5.11 Multiple Devices with the same IE

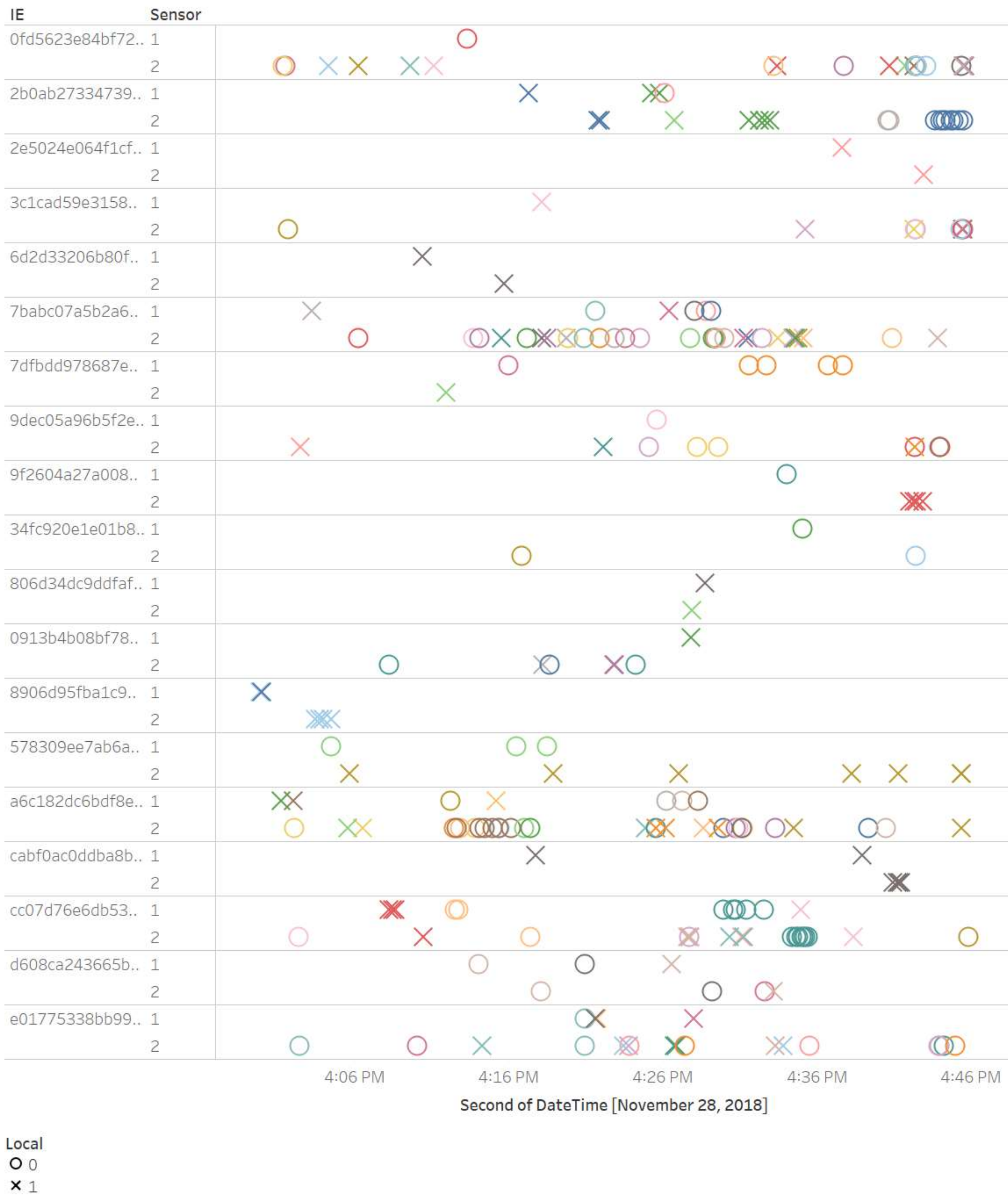

Figure 5.12 IEs captured by different sensors

Figure 5.12 shows 19 unique IEs that have been captured by both sensors. In the figure, different colors represent different MAC addresses, and the shapes represent whether the IEs are associated with locally assigned MAC address. Detailed data of Figure 5.12 is attached in Appendix A.

According to the Figure, although 13 of 19 IEs have associated with multiple devices, which indicates that IEs are more likely to be the signature of device module and chipset rather than individual devices, about 5% extra trips could still be recognized by IEs in a proper timing window. In addition, 6+ trips have been recognized by local MAC addresses, as some randomized addresses don't change in a 5-minutes time period. It reveals that the short-distance travel time measurement is less affected by the MAC address randomization.

In a word, IEs are not good enough for recognizing individual devices independently. However, combined with the MAC addresses, proper threshold and timing window, it would increase the trip sampling rates to some extent.

5.7 OTHER TECHNIQUES

Besides Information Elements, there are some techniques which have been discovered by other researchers to identify devices with randomized MAC-addresses. The Universally Unique Identifier-Enrollee (UUID-E), which is a field in Wi-Fi Protected Setup (WPS) IE, can be used to retrieve global MAC addresses[22]. However, most of the probe request frames in our experiment doesn't contain the WPS IE fields. The predictable scrambles seeds in Wi-Fi physical layer is another field could be used to link multiple random MAC addresses to the same devices[22], [78], but it requires additional devices such as software-defined radio to receive and decode Wi-Fi signals.

In addition, as all the mobile devices only perform MAC address randomization at the discovery stages, the hardware MAC addresses can be used to connect the hotspot. Plus, mobile devices also automatically connect to the Wi-Fi hotspot which has been connected and recognized by the devices in the past. By installing a free Wi-Fi hotspot network with multiple routers, all the devices connected to the hotspot could be observed and tracked. This data collection strategy has been majorly used in the shopping mall to find user's preference and shopping pattern, which also have the potential to be applied in the transportation area.

5.8 SUMMARY

In this chapter, we introduce MAC randomization mechanisms which vary between operating systems and the model of network cards, evaluate the impact of MAC address randomization strategies on the MAC-address based travel time data collection. As MAC address randomization makes devices anonymous, it breaks the assumption and of the traditional MAC-based device tracking methodology. Two criteria, the trip sampling rate and the collision rate are computed to measure the effectiveness of MAC data collection and trip recognition. With the increasing amounts of randomized devices, trip sampling rate is decreasing correspondingly, which will have potential impacts on the reliability of mobile device sensing technology as the accuracy of travel time estimation is correlated with the sample size. For mismatching problems, the probability of MAC collision that shares the same prefix is low. Since many devices didn't implement MAC randomization, the MAC address mismatching will not have large effects on the travel time accuracy. Using Information Elements combined with MAC address to recognized trips have limited effects on increasing trip sampling rate, but there is a high probability that different devices have the same Information Elements signature, the effectiveness depends highly on the scenario.

Re-identifying mobile devices with randomized MAC address is a big challenge for travel time measurement applications since the detection time window is too short for each vehicle, cyclist or pedestrian, ranging from several seconds to several minutes. In such a short time period, only limited probe request frames could be captured. Moreover, a device won't be tracked in the system until it is detected by the next sensor, and some detections could be missed. Because of the blind time of each device, sequence number-based or timing pattern-based tracking methods could not be applied. Other techniques, such as the predictable scrambles seeds in the physical layer and Wi-Fi hotspot tracking, may have the ability to mitigate the impact, which asks for future experiments and evaluations in this topic.

Chapter 6. CONCLUSION

This study reviews MAC-address based travel time measurement techniques and evaluates the effectiveness of different filtering algorithms to improve the travel time measurement accuracy. A mobile device sensing data collection and processing framework is proposed. The recent challenges due to MAC address randomization mechanisms are analyzed with several experiments. This paper provides a comprehensive understanding of the principle, practice, and challenges in MAC address-based travel time measurement techniques.

For travel time measurement technology, several major challenges, as MAC address matching due to multiple detections, noises filtering due to different travel modes, and fluctuations due to low samples have been discussed. The travel time result is calculated by Acyclica data in Seattle and is compared with the ground truth travel time calculated by the license plate reader data. For multiple MAC detection matching issue, RSSI-value based matching strategy is more applicable for uncontrolled traffic, while time order-based strategy is suitable for signalized intersections considering of either upstream or downstream intersection delay. Since the travel time paired by the raw MAC addresses has lots of noises, proper filtering algorithms are required to smooth the travel time result. A density-based clustering DBSCAN algorithm is applied to remove travel time outliers. Compared to other previous algorithms, including Moving Median Filtering, Moving Median Absolute Deviation filtering, Box-and-whisker filtering, DBSCAN has reached the most accurate travel time result with the minimum deviation from LPR data. To sum up, MAC address based mobile device sensing which utilizes the wireless frame from mobile devices can effectively measure the traffic on the road. Proper pairing and filtering algorithms are required to reduce the noise to improve travel time measurement accuracy. In addition, the number of valid samples paired by MAC address is one of the essential factors which affects the accuracy of travel time measurement.

Based on the findings in the travel time measurement algorithms, a mobile device sensing data collection and processing framework is proposed. As a client-server framework with multiple independence modules and components, the framework is scalable, robustness and able to downgrade. In addition, the framework could be extended and customized based on the usage situation, can be applied for most of the scenarios to process mobile device sensing data for travel time measurement and transportation data collection needs.

As MAC address randomization was developed in 2014 and enhanced during the recent years, there were no researches in the transportation area had investigated the potential impact on mobile device sensing data collection. This research reviews recent papers on MAC address randomization and raises two criteria, trip sampling rate and collision rate, to measure the influence of the new changes. With the increasing amounts of randomized devices, the trip sampling rate is decreasing correspondingly which results in less valid samples paired by MAC address. The decrease in the trip sampling rate would have risks on the reliability of MAC address-based travel time measurement in the future, since the number of valid samples is one of the essential factors contribute to the accuracy of measurement. The probability of MAC collision is low that different devices have less chance to randomize the same MAC address. Besides MAC address, other indicators in the probe request frames, such as Information Elements are not enough to distinguish and recognize different devices. Other techniques, such as the predictable scrambles seeds in the physical layer and Wi-Fi hotspot tracking, may have the ability to mitigate the impact, which asks for future experiments and evaluations in this topic.

BIBLIOGRAPHY

[1] W. Chung, C. Yau, K. Shin, and R. Myllyla, “A Cell Phone Based Health Monitoring System with Self Analysis Processor using Wireless Sensor Network Technology,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 3705–3708.

[2] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, “Real-time Air Quality Monitoring Through Mobile Sensing in Metropolitan Areas,” in *Proceedings of the 2Nd ACM SIGKDD International Workshop on Urban Computing*, New York, NY, USA, 2013, pp. 15:1–15:8.

[3] D.-S. Han, I.-J. Lee, and J.-Y. KOO, “Method of estimating location of mobile device in transportation using WiFi,” US8594003B2, 26-Nov-2013.

[4] “Travel mode identification using bluetooth technology: Journal of Intelligent Transportation Systems: Vol 22, No 5.” [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/15472450.2017.1384698. [Accessed: 06-Nov-2018].

[5] J. A. Burke *et al.*, “Participatory sensing,” May 2006.

[6] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, “A survey of mobile phone sensing,” *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010.

[7] O. Järv, R. Ahas, and F. Witlox, “Understanding monthly variability in human activity spaces: A twelve-month study using mobile phone call detail records,” *Transp. Res. Part C Emerg. Technol.*, vol. 38, pp. 122–135, Jan. 2014.

[8] H. Wang, F. Calabrese, G. D. Lorenzo, and C. Ratti, “Transportation mode inference from anonymized and aggregated mobile phone call detail records,” in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 318–323.

[9] J. Wu *et al.*, “Traffic flow simulation based on call detail records,” *Acta Geogr. Sin.*, vol. 12, no. 67, pp. 1657–1665, 2012.

[10] F. Wang and C. Chen, "On data processing required to derive mobility patterns from passively-generated mobile phone data," *Transp. Res. Part C Emerg. Technol.*, vol. 87, pp. 58–74, Feb. 2018.

[11] F. Wang and C. Chen, "Passively-Generated Mobile Phone Data: Data Issues and Processing Required," presented at the Transportation Research Board 97th Annual MeetingTransportation Research Board, 2018.

[12] J. S. Wasson, J. R. Sturdevant, and D. M. Bullock, "Real-Time Travel Time Estimates Using Media Access Control Address Matching," *Inst. Transp. Eng. ITE J. Wash.*, vol. 78, no. 6, pp. 20–23, Jun. 2008.

[13] N. Abedi, A. Bhaskar, and E. Chung, "Bluetooth and Wi-Fi MAC Address Based Crowd Data Collection and Monitoring: Benefits, Challenges and Enhancement," p. 17.

[14] A. Haghani, M. Hamedi, K. Sadabadi, S. Young, and P. Tarnoff, "Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2160, pp. 60–68, Sep. 2010.

[15] Y. Wang, Y. Malinovskiy, Y.-J. Wu, and U. K. Lee, "Error Modeling and Analysis for Travel Time Data Obtained," p. 83.

[16] A. Saeedi, S. Park, D. S. Kim, and J. D. Porter, "Improving Accuracy and Precision of Travel Time Samples Collected at Signalized Arterial Roads with Bluetooth Sensors," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2380, no. 1, pp. 90–98, Jan. 2013.

[17] S. M. Quayle, P. Koonce, D. Depencier, and D. M. Bullock, "Arterial Performance Measures with Media Access Control Readers: Portland, Oregon, Pilot Study," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2192, no. 1, pp. 185–193, Jan. 2010.

[18] N. Abedi, A. Bhaskar, E. Chung, and M. Miska, "Assessment of antenna characteristic effects on pedestrian and cyclists travel-time estimation based on Bluetooth and WiFi MAC addresses," *Transp. Res. Part C Emerg. Technol.*, vol. 60, pp. 124–141, Nov. 2015.

[19] M. Cunche, “I know your MAC address: targeted tracking of individual using Wi-Fi,” *J. Comput. Virol. Hacking Tech.*, vol. 10, no. 4, pp. 219–227, Nov. 2014.

[20] L. Hutchinson, “iOS 8 to stymie trackers and marketers with MAC address randomization,” *Ars Tech.*, 2014.

[21] J. Martin *et al.*, “A Study of MAC Address Randomization in Mobile Devices and When it Fails,” *Proc. Priv. Enhancing Technol.*, vol. 2017, no. 4, pp. 365–383, Oct. 2017.

[22] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, “Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms,” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security - ASIA CCS ’16*, Xi’an, China, 2016, pp. 413–424.

[23] C. Matte, “Wi-Fi Tracking: Fingerprinting Attacks and Counter-Measures,” Theses, Université de Lyon, 2017.

[24] S. Turner, “Travel Time Data Collection Handbook,” p. 341.

[25] C. de Fabritiis, R. Ragona, and G. Valenti, “Traffic Estimation And Prediction Based On Real Time Floating Car Data,” in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 197–203.

[26] S. M. Turner, “Advanced Techniques for Travel Time Data Collection,” *Transp. Res. Rec.*, p. 8.

[27] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*. Springer Science & Business Media, 2012.

[28] R.-P. Schäfer, K.-U. Thiessenhusen, and P. Wagner, “A TRAFFIC INFORMATION SYSTEM BY MEANS OF REAL-TIME FLOATING-CAR DATA,” p. 8.

[29] INRIX, “Traffic Data from INRIX Shows You What’s Happening Down the Road,” *INRIX - INRIX*. [Online]. Available: http://inrix.com/blog/2017/05/look-a-mile-ahead-how-inrix-

informs-people-and-machines-about-whats-happening-in-front-of-them/. [Accessed: 17-Nov-2018].

[30] INRIX, "INRIX Traffic Brochure - INRIX," *INRIX*. [Online]. Available: http://inrix.com/resources/inrix-traffic-brochure/. [Accessed: 17-Nov-2018].

[31] W. Pu, J. (Jane) Lin, and L. Long, "Real-Time Estimation of Urban Street Segment Travel Time Using Buses as Speed Probes," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2129, no. 1, pp. 81–89, Jan. 2009.

[32] X. Liu, S. Chien, and K. Kim, "Evaluation of floating car technologies for travel time estimation," *J. Mod. Transp.*, vol. 20, no. 1, pp. 49–56, Mar. 2012.

[33] V. Wieser, J. Duha, M. Bahleda, R. Odrobiňák, and D. Grendár, "Application of Positioning Systems in Intelligent Transport Systems," Jan. 2003.

[34] C. Patel, D. Shah, and A. Patel, "Automatic Number Plate Recognition System (ANPR): A Survey," *Int. J. Comput. Appl.*, vol. 69, no. 9, pp. 21–33, May 2013.

[35] M. D. Bailey and F. G. Rawling, *A Computerized Travel Time Study for Northeastern Illinois: Methodology and Commentary*, vol. 91. Chicago Area Transportation Study, 1991.

[36] S. Turner, "License Plate Matching Techniques," p. 42.

[37] N. R. Velaga *et al.*, "Development of a Map-Matching Algorithm for Rural Passenger Information Systems through Mobile Phones and Crowd Sourcing," *J. Comput. Civ. Eng.*, vol. 27, no. 6, pp. 732–742, Nov. 2013.

[38] J. L. Gifford, L. Yermack, and C. A. Owens, "E-ZPass: Case Study of Institutional and Organizational Issues in Technology Standards Development," *Transp. Res. Rec.*, p. 5.

[39] M. Hirose, "Newly Obtained Records Reveal Extensive Monitoring of E-ZPass Tags Throughout New York," *American Civil Liberties Union*. [Online]. Available: https://www.aclu.org/blog/privacy-technology/location-tracking/newly-obtained-records-reveal-extensive-monitoring-e-zpass. [Accessed: 17-Nov-2018].

[40] E. Niver, K. C. Mouskos, T. Batz, and P. Dwyer, “Evaluation of the TRANSCOM’s system for managing incidents and traffic (TRANSMIT),” *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 15–31, Mar. 2000.

[41] A. Bhaskar and E. Chung, “Fundamental understanding on the use of Bluetooth scanner as a complementary transport data,” *Transp. Res. Part C Emerg. Technol.*, vol. 37, pp. 42–72, Dec. 2013.

[42] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, “Precise Indoor Localization Using Smart Phones,” in *Proceedings of the 18th ACM International Conference on Multimedia*, New York, NY, USA, 2010, pp. 787–790.

[43] T. Oransirikul, R. Nishide, I. Piumarta, and H. Takada, “Measuring Bus Passenger Load by Monitoring Wi-Fi Transmissions from Mobile Devices,” *Procedia Technol.*, vol. 18, pp. 120–125, Jan. 2014.

[44] S. El-Tawab, R. Oram, M. Garcia, C. Johns, and B. B. Park, “Poster: Monitoring transit systems using low cost WiFi technology,” in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016, pp. 1–2.

[45] N. Shlayan, A. Kurkcu, and K. Ozbay, “Exploring pedestrian Bluetooth and WiFi detection at public transportation terminals.,” in *ITSC*, 2016, pp. 229–234.

[46] Y. Malinovskiy and Y. Wang, “Pedestrian Travel Pattern Discovery Using Mobile Bluetooth Sensors,” presented at the Transportation Research Board 91st Annual MeetingTransportation Research Board, 2012.

[47] M. Blogg, C. Semler, M. Hingorani, and R. Troutbeck, “Travel Time and Origin-Destination Data Collection using Bluetooth MAC Address Readers,” p. 15, 2010.

[48] “IEEE 802.11-2016 - IEEE Standard for Information technology?Telecommunications and information exchange between systems Local and metropolitan area networks?Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.” [Online]. Available: https://standards.ieee.org/standard/802_11-2016.html. [Accessed: 24-Oct-2018].

[49] E. Ferro and F. Potorti, “Bluetooth and Wi-Fi wireless protocols: a survey and a comparison,” *IEEE Wirel. Commun.*, vol. 12, no. 1, pp. 12–26, Feb. 2005.

[50] T. Tsubota, A. Bhaskar, E. Chung, and R. Billot, “Arterial traffic congestion analysis using Bluetooth Duration data,” p. 14, 2011.

[51] L. M. Kieu, A. Bhaskar, and E. Chung, “Bus and car travel time on urban networks: integrating bluetooth and bus vehicle identification data,” 2012.

[52] S. Yang and Y.-J. Wu, “Travel mode identification using bluetooth technology,” *J. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 407–421, Sep. 2018.

[53] “Traffic Flow Map Volumes | City of Seattle Open Data portal.” [Online]. Available: https://data.seattle.gov/Transportation/Traffic-Flow-Map-Volumes/38vd-gytv. [Accessed: 02-Nov-2018].

[54] K. Douglas and S. Douglas, *PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgresSQL Databases*. Sams Publishing, 2003.

[55] B. I. Justusson, “Median Filtering: Statistical Properties,” in *Two-Dimensional Digital Signal Prcessing II: Transforms and Median Filters*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 161–196.

[56] M. Ester, H.-P. Kriegel, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” p. 6.

[57] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson Correlation Coefficient,” in *Noise Reduction in Speech Processing*, I. Cohen, Y. Huang, J. Chen, and J. Benesty, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4.

[58] J. D. Brock, R. F. Bruce, and M. E. Cameron, “Changing the world with a Raspberry Pi,” *J. Comput. Sci. Coll.*, vol. 29, no. 2, pp. 151–153, 2013.

[59] W. She-min, “Extremely accurate real-time clock chip DS3231 and its application in multi-rate Watt-hour meter [J],” *Electr. Meas. Instrum.*, vol. 2, p. 019, 2007.

[60] J. Postel, “User datagram protocol,” 1980.

[61] W. Richard Stevens, “TCP/IP Illustrated, volume 1, The Protocols “, Addision-Wesley,” *Read. Mass.*, 1994.

[62] K. Thompson, “Programming techniques: Regular expression search algorithm,” *Commun. ACM*, vol. 11, no. 6, pp. 419–422, 1968.

[63] M. Stonebraker and G. Kemnitz, “The POSTGRES next generation database management system,” *Commun. ACM*, vol. 34, no. 10, pp. 78–92, 1991.

[64] P. Ramsey, “Postgis manual,” *Refract. Res. Inc*, 2005.

[65] E. Naiburg, E. J. Naiburg, and R. A. Maksimchuck, *UML for database design*. Addison-Wesley Professional, 2001.

[66] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, “Anonysense: privacy-aware people-centric sensing,” in *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys ’08*, Breckenridge, CO, USA, 2008, p. 211.

[67] K. Skinner and J. Novak, “Privacy and your app,” in *apple worldwide dev. conf.(wwdc)*, 2015.

[68] “Android 6.0 Changes,” *Android Developers*. [Online]. Available: https://developer.android.com/about/versions/marshmallow/android-6.0-changes. [Accessed: 21-Oct-2018].

[69] C. Huitema, “Experience with MAC Address Randomization in Windows 10,” p. 6.

[70] “Analysis of iOS 8 MAC Randomization on Locationing,” p. 6.

[71] I. S. Association, “Guidelines for Use of Organizationally Unique Identfiier (OUI) and Company ID (CID),” *IEEE Stand. Assoc.*, 2015.

[72] D. Wagner, “A Generalized Birthday Problem,” in *Advances in Cryptology — CRYPTO 2002*, 2002, pp. 288–304.

[73] “IDC : Huawei continue to lead in China smartphone sales in 3rd quarter of 2017,” *GadgetDetail*, 08-Nov-2017. [Online]. Available: http://www.gadgetdetail.com/idc-huawei-continue-lead-china-smartphone-sales-3rd-quarter-2017/. [Accessed: 22-Oct-2018].

[74] M. Gast, *802.11 Wireless Networks: The Definitive Guide*. O’Reilly Media, Inc., 2005.

[75] “9 best packet sniffers and network analyzers for 2018,” *Comparitech*, 19-Jul-2018. .

[76] U. Lamping, R. Sharpe, E. Warnicke, U. Lamping, R. Sharpe, and E. Warnicke, *Wireshark User’s Guide For Wireshark 2.1*. 2014.

[77] C. Matte and M. Cunche, “DEMO: Panoptiphone: How Unique is Your Wi-Fi Device?,” in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, New York, NY, USA, 2016, pp. 209–211.

APPENDIX A

Attached table shows the result of using Information Elements to recognized device. SENSOR represents the sensor ID, local bit represents whether the hashed MAC address is a local address.

SENSOR	Hashed_IE	LOCAL_BIt	HASHED_MAC	DATETIME
1	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	f7d22a9fd768074ed8d21814f5d5cb3d	11/28/2018 16:25
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	38af45c48e520e062504565fd7520dff	11/28/2018 16:02
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	cc636afcab4260b911061baf1e322e9c	11/28/2018 16:22
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	631352c5abac19c92100fd885c1f938b	11/28/2018 16:25
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	de837834a7ae56673f7d6eab8147f6f9	11/28/2018 16:28
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	83e50840870bed120ba76d8031a93f27	11/28/2018 16:29
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	e3a2e59c2c2eec59c150a8b00587b413	11/28/2018 16:42
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	483e97a4c119e37382169e840879f98e	11/28/2018 16:42
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	1babb47f320b2e5ea5e3aeee8588499c	11/28/2018 16:43
2	9dec05a96b5f2e2803b39c834d886d8d33110ffc6d07a362380fdbf32a64f103	1	efcfc32a63f1b76b5cccd492cb9be443	11/28/2018 16:43
1	7dfbdd978687e00531f573fe073eb8d83c63b92ce4bf9c0552140d58ae2b9d23	0	6e87f502ebaf45e2599578f50dd222d5	11/28/2018 16:15
1	7dfbdd978687e00531f573fe073eb8d83c63b92ce4bf9c0552140d58ae2b9d23	0	f55cdb2c4d87582e5bce75523d229beb	11/28/2018 16:31
2	7dfbdd978687e00531f573fe073eb8d83c63b92ce4bf9c0552140d58ae2b9d23	0	bc48e959a927d5c5d881fdab73723ab5	11/28/2018 16:11
1	6d2d33206b80f0ad16a025876c86dac8106288205fe70316d1a87fdf0e001997	0	a8f49792e7b4acee7aedf3b9551cb34f	11/28/2018 16:10
2	6d2d33206b80f0ad16a025876c86dac8106288205fe70316d1a87fdf0e001997	0	a8f49792e7b4acee7aedf3b9551cb34f	11/28/2018 16:15
1	9f2604a27a008ead2af7d8778ed05e285d3baccaf984ca5279860751041f85ab	0	6415ce7557397997290ad7046a8486c8	11/28/2018 16:34
2	9f2604a27a008ead2af7d8778ed05e285d3baccaf984ca5279860751041f85ab	0	414f1af34e6cae294102594f05fc9d54	11/28/2018 16:42
1	578309ee7ab6ada2d3df8d78da86c8d0955ae9e6f20313f15fcb092068438be8	0	5ea98d0b8cab0359ed5114260695f393	11/28/2018 16:04

2	578309ee7ab6ada2d3df8d78da86c8d0955ae9e6f20313f15fcb092068438be8	0	3cef59b550d21ca44e7c218e84d90989	11/28/2018 16:05
1	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	7a18c5333452189ac807bf590171a369	11/28/2018 16:14
1	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	ffa7705a0a14d2dae010bbfa2bfe6035	11/28/2018 16:20
1	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	67e366c867b81f2b91ce6fee167ff1e4	11/28/2018 16:26
2	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	7a18c5333452189ac807bf590171a369	11/28/2018 16:18
2	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	ffa7705a0a14d2dae010bbfa2bfe6035	11/28/2018 16:29
2	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	4119140dfcf332c347fd3dbd7292479a	11/28/2018 16:32
2	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	67e366c867b81f2b91ce6fee167ff1e4	11/28/2018 16:33
2	d608ca243665b6ffc440ee1baaf5a0abde41c5168b5897fc5995bd0bc67db666	0	30bd0fb02b9300c6d7941622d8214734	11/28/2018 16:47
1	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	91f500e56efc03509bef15b3db8efd33	11/28/2018 16:08
1	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	7a74d566a4e413953f0562b239374c92	11/28/2018 16:12
1	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	c0545474340de7f64debe72127b924e2	11/28/2018 16:29
1	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	9533d19442880622818f81a00e349ca9	11/28/2018 16:34
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	92a502ee23a285c230ab15d8c84652b6	11/28/2018 16:02
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	91f500e56efc03509bef15b3db8efd33	11/28/2018 16:10
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	7a74d566a4e413953f0562b239374c92	11/28/2018 16:17
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	3f47ca1569a991539f7955ef0c46dba2	11/28/2018 16:27
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	adb32ab955a7b489426daa0df76d404a	11/28/2018 16:27
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	d0c9029d34f5a597cc25d339a33b0259	11/28/2018 16:30
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	38eb2cbb1c498a4f14314f0b388170ca	11/28/2018 16:31
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	c0545474340de7f64debe72127b924e2	11/28/2018 16:34
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	9533d19442880622818f81a00e349ca9	11/28/2018 16:38
2	cc07d76e6db5345cf36288d211c8fcea0462641927081e270ffb0129c20b4c74	0	f42e873be01bc0617e3546e33275568e	11/28/2018 16:45

1	34fc920e1e01b8cad5fba3ce2f89a5fac1 03ebe6647c10d909f15db03322e0ba	0	f76b240305d79edcb 4bfedb4b3683e01	11/28/2018 16:35
2	34fc920e1e01b8cad5fba3ce2f89a5fac1 03ebe6647c10d909f15db03322e0ba	0	a190b28b983891fea dbd1c4e84110b91	11/28/2018 16:16
2	34fc920e1e01b8cad5fba3ce2f89a5fac1 03ebe6647c10d909f15db03322e0ba	1	411979ed31b4848a 4de348ac99402053	11/28/2018 16:42
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	f1023559b34800e03 4f775c7d55908a8	11/28/2018 16:01
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	1fb12567bc7a80e0b 0bb0ce47e640836	11/28/2018 16:02
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	0a68da457ccfe1909 3050a301b9dc918	11/28/2018 16:12
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	27830e9d67cae990 7b7da49fa6c69418	11/28/2018 16:15
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	1	00cf825e39f572d6a 99853f6ee7aca59	11/28/2018 16:26
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	1	c7282b1d7c1f2b87f 4079d2eeb637b09	11/28/2018 16:27
1	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	1	4de58f746823307f2 1200228b241ba2d	11/28/2018 16:28
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	cee32635cf8d1d2e3 917eab32209f25e	11/28/2018 16:02
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	39f4bffec625cac51c 880344ca34176c	11/28/2018 16:05
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	23a44b21b5b44f1c1 46e3d8b14ecd10d	11/28/2018 16:06
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	dbfdac864ede45ceb 9075bffffff3e4d	11/28/2018 16:12
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	224925105ab94354 3de6c93790980407	11/28/2018 16:12
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	1216923017948e61 77c96979bd23918c	11/28/2018 16:12
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	49ffabdc8f7eb9d7b 46d9fe90b0323ce	11/28/2018 16:13
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	6c5e5c316022e1e46 d93d4481c32b068	11/28/2018 16:14
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	07b8b3e9730e7b87 9c7cca9a1948d599	11/28/2018 16:15
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	72f1b202b1a75f160 7af0960a5e5e25c	11/28/2018 16:17
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	3e89bbc1c00a5b5d 3f083e00a9179ddd	11/28/2018 16:17
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	7c882b2ec63f7c619 affb25fca140193	11/28/2018 16:24
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d 7a42a10cb7c617e38a9472fd7e4e87e	0	980d761b2194edfe 6f99a6ef7f1f2a6c	11/28/2018 16:25

2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	7634b919ac18aef45d13781836aaaf28	11/28/2018 16:25
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	ffded43c2f2fc9c7d1007f52077b0353	11/28/2018 16:28
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	eaf2f152b05943b72c12b6c9b0c0c403	11/28/2018 16:29
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	1	c8fbd1519129fe5fe3c404fa16f3a2cb	11/28/2018 16:29
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	1	1f7204f6aa204dc2a2bc0019bd0b5463	11/28/2018 16:30
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	1	b9c371c1d1e8d7228e9d1cd5b800576a	11/28/2018 16:30
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	1	420a26f636e7fc1a124594b68ef1077a	11/28/2018 16:31
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	abd8f052c7df08e61d6d4ff49fa3bc23	11/28/2018 16:31
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	1b0a98a96e4f4b1879adf4edc774d672	11/28/2018 16:33
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	4bc218c67ddc43e35bfb5d7bf27d9377	11/28/2018 16:34
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	c444bd60bee2271b1dcd9ccf56e0b4d1	11/28/2018 16:39
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	b200558930e2daceb69e7e7f31e255f3	11/28/2018 16:40
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	4c35b16cabae3e41ddcf0ba1ab127e62	11/28/2018 16:45
2	a6c182dc6bdf8ee21efa7204f0dfb2a9d7a42a10cb7c617e38a9472fd7e4e87e	0	8c0fb8bdca70be63a22d94735e3247ee	11/28/2018 16:47
1	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	a8f3ad544436a8f557b6bb6fae3241cd	11/28/2018 16:20
1	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	1e9a5fbf4e8b5d3cf68242e0c5e8d063	11/28/2018 16:21
1	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	f07100dc3394a128ec033e92fefaa16e	11/28/2018 16:21
1	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	561d3e32760a2dc0d863d2a7761a9b50	11/28/2018 16:27
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	9a4b0a4e7a20e2231e6a6d3cec3b6b35	11/28/2018 16:02
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	880cd6b76424e742e7ef4e05df866c4a	11/28/2018 16:10
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	d9386afe4efdb18aa19ade10acfa1e76	11/28/2018 16:14
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	a8f3ad544436a8f557b6bb6fae3241cd	11/28/2018 16:20
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	fbb1b75bb57a17ba3ee69954f09f9cda	11/28/2018 16:23

2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	bb007eb606c1480ddc8a01c5839c91c2	11/28/2018 16:23
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	a04e6c9a6147c1772409de27e823ddc7	11/28/2018 16:23
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	7674169a79723741aff04c3fb1bd89d0	11/28/2018 16:26
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	1e13d1d73d9cb0735670ecab35566089	11/28/2018 16:26
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	74087b9a1a6499b5832902d14b31ac4e	11/28/2018 16:27
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	523ee2da601c95424050135bf1779869	11/28/2018 16:33
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	d373b9f27a07255f1286767ae228630a	11/28/2018 16:33
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	dd4ce52f36fe8e5c96118bda13649943	11/28/2018 16:35
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	a0bf62e88082b8fd4948b8db258315f1	11/28/2018 16:43
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	fba368d8a9a557c3e214728c59823b10	11/28/2018 16:43
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	23a02060b5db0e3ceb3126941e43b2f9	11/28/2018 16:44
2	e01775338bb99d6ed11a05c4b647bd694843fa79ba5a82063dbafacfc9e0368c	1	03741a6ce5d16d15cf67ca356888459b	11/28/2018 16:44
1	8906d95fba1c95dcc7c37c80289c2b39b44538f37da6d53208e2825eae28f897	0	e21bc9996eb0ca11a799119cade0de20	11/28/2018 15:59
1	8906d95fba1c95dcc7c37c80289c2b39b44538f37da6d53208e2825eae28f897	0	8cebd367d8669de4790bfa4518bb8282	11/28/2018 15:59
2	8906d95fba1c95dcc7c37c80289c2b39b44538f37da6d53208e2825eae28f897	0	e21bc9996eb0ca11a799119cade0de20	11/28/2018 16:03
1	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	ab53c78ddf16f35b52786a9c88c3dccd	11/28/2018 16:13
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	85cc0854b73d518a8135cb0cc1b7d952	11/28/2018 16:01
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	21d1130da41802f67c805a0d87c2e688	11/28/2018 16:01
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	a3b465fd0fc32efa26051523272c71c0	11/28/2018 16:04
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	d2bbdc591bacbb90132c2a97a6d04cab	11/28/2018 16:06
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	27a05420e44ea8b563ec01993db887af	11/28/2018 16:09
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	97fc511f0bd64218a065827b7c73fb89	11/28/2018 16:11
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	7688dba636289f9d9bbca8161ba73a51	11/28/2018 16:33

2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	6ec291a9e7f5574273e06db230f45394	11/28/2018 16:33
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	27e169f399489be3aef0eb20ae40e91a	11/28/2018 16:37
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	458fcd013b271a4d42957f7252e1710c	11/28/2018 16:40
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	129cf36229a74caa0239408bbb24cce6	11/28/2018 16:41
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	befea927995ea1bf7285eaee1eb77a4d	11/28/2018 16:42
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	3fe1e36f67afe435cbbdc738a3436558	11/28/2018 16:42
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	dba2326eb29d891405aa2926af2674ee	11/28/2018 16:42
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	15435b85720433cb0e26bcf073a3e911	11/28/2018 16:42
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	fb6df1691372f510e6264954135931b1	11/28/2018 16:43
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	423c259001a853ccdcf843520001cb59	11/28/2018 16:45
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	fba1fad1352aabc39ebcab82cb2339a0	11/28/2018 16:45
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	de238a08ffe22acbe2f3e11fcffede29	11/28/2018 16:45
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	2975e41500131af40e37deec758e1ad1	11/28/2018 16:45
2	0fd5623e84bf72b247dd7e3acaf2f16f188a7b5242ff673666ddf772673ae07b	1	59f8fc22ae59e8da6e82dddecff93843	11/28/2018 16:45
1	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	cb8af6170db349736b2e726d9762bcc5	11/28/2018 15:59
1	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	63fdaa345f06052b59ce9e6f4bc6e1b2	11/28/2018 16:27
2	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	2559eba92fd9a3ac4671f2c2eb726aa8	11/28/2018 16:08
2	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	f7fd24bd69741f7e1096a0b21afffe81	11/28/2018 16:18
2	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	971ee7d33739573c5cc0d34e86f44e8c	11/28/2018 16:18
2	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	d38a5851f83e83fd8487c7ee6bcacd3b	11/28/2018 16:22
2	0913b4b08bf7830221c8907fa5aff483a4d7e671e154c92a16d20b42d30019c5	1	6054afe2cde071d137138132e16cdb8f	11/28/2018 16:24
1	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	73bd3af71d615281ab9ea9f26b32835b	11/28/2018 16:03
1	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	910415488a962facc6d7a3a088a2a75d	11/28/2018 16:21

1	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	2fc08f1d1bec0b5e5 bd14c87aab3235d	11/28/2018 16:26
1	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	80cbee5668555bed d0ade81cb95227d6	11/28/2018 16:28
1	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	b0865044460b4039 bd918fad6e888b28	11/28/2018 16:28
1	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	9444770815de8194 7fdf0cb7e3097319	11/28/2018 16:29
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	c61c3439b0575cca9 ba68195b9e8e68c	11/28/2018 16:06
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	c22370a39505d22e c0be2cf23e2aab18	11/28/2018 16:13
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	2ca4257408618aa0 3b6c0dc7576515ff	11/28/2018 16:14
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	bb481de4fc9b57ca6 10908ede30c4d54	11/28/2018 16:15
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	b99538d94b087a7d 85716c4d27f609b1	11/28/2018 16:17
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	b0d440e34fa22b75 87ccb4a001795046	11/28/2018 16:18
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	f24752d71482ff67f0 cee98d0865f769	11/28/2018 16:18
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	930dbaaf75e699f7c 2d8a398a2243161	11/28/2018 16:19
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	09b5db8d28e4633b ad30e5b093521633	11/28/2018 16:19
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	d567acf839aeba413 c169cd2006f4b44	11/28/2018 16:20
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	f9b93366d460de8fb e9b00615ce75db4	11/28/2018 16:21
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	83453e077d05989d a64561609f587a8c	11/28/2018 16:22
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	8cf60ff8918121bfbe 57d5dfaa8e373d	11/28/2018 16:23
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	d3b77c99c76940ba 0547e2bb8ffbad39	11/28/2018 16:24
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	46b477777edbdecb c88f7199b19e919a	11/28/2018 16:27
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	0	1a0a0a090bfd973bb 6d5370e5ba32daa	11/28/2018 16:29
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	1ff26543a931b6a44 b7b1e7cd9c5cc12	11/28/2018 16:29
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	8d763b470259096b da942a675ab2ffdf	11/28/2018 16:29
2	7babc07a5b2a6b577c201afb2551565b 766f55abd90d24f134a39c9e35c52166	1	6e4ebd3dea7328b1 77592260b228d6ba	11/28/2018 16:31

2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	3c09d97d897e7ad2ec962ca2d241e7bc	11/28/2018 16:31
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	6a02bbaa85b9bf1f2b67f6b6917452ef	11/28/2018 16:32
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	3f532eff2407d399a89eab599d130e38	11/28/2018 16:33
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	2a0ec8b4799fabfcc59f0861a00e3fd4	11/28/2018 16:34
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	704eba428f51ac9ce92ac7c8bf988a23	11/28/2018 16:34
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	53585c776d0fcdf2d633390d5bae4b65	11/28/2018 16:34
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	2ed4e6b96b2b279fac93dd24a7056d4f	11/28/2018 16:34
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	c1c39968cfd63b7ca2e1791500e572b1	11/28/2018 16:35
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	3574daab207c9975124b710027c4d91e	11/28/2018 16:40
2	7babc07a5b2a6b577c201afb2551565b766f55abd90d24f134a39c9e35c52166	1	3c34425f5fd2e2a84eadfaed2c4f2b40	11/28/2018 16:43
1	2e5024e064f1cf3497c7a6cd7d8052106fd4cf3034979a038646b3da7303770a	0	bed94ebb3bf4335f8d9a5f8e3e13fc64	11/28/2018 16:37
2	2e5024e064f1cf3497c7a6cd7d8052106fd4cf3034979a038646b3da7303770a	0	bed94ebb3bf4335f8d9a5f8e3e13fc64	11/28/2018 16:42
1	cabf0ac0ddba8b091e2c6c908784316e0a9d6ead89f24d8bd0fc1c869c0c24b7	0	1d400b1fc23b2f688760fc890b571c7b	11/28/2018 16:17
2	cabf0ac0ddba8b091e2c6c908784316e0a9d6ead89f24d8bd0fc1c869c0c24b7	0	1d400b1fc23b2f688760fc890b571c7b	11/28/2018 16:40
1	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	2c51d3b7ea9d8f4856414c168afae374	11/28/2018 16:18
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	8891d4228af57281a2d35dc634156ce0	11/28/2018 16:01
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	7a640bb146beb5096dea7beaec4f311a	11/28/2018 16:35
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	66f788731f2e2ab4615c492c2bb4442e	11/28/2018 16:42
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	3b3f5dd1979dfbcf0784ed103cb26fda	11/28/2018 16:42
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	0ebdaf53b8b58f8ca4464c49d7aa0a38	11/28/2018 16:42
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	baf6e6e27787459fc142e9c2b5d399c7	11/28/2018 16:45
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	07a139ba7d24b4dc5a223741d88c2814	11/28/2018 16:45
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	340187b845a84447acdacca11e1cd1a4	11/28/2018 16:45

2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	4376ddd67373c43c2694baa5d9614301	11/28/2018 16:45
2	3c1cad59e3158bf38c1276a480c4e84e6a082adb799103119006331239c2fa44	1	e1e07b9801e7d65ebd352750114a179a	11/28/2018 16:45
1	806d34dc9ddfafe3a9346a6b268daa1d42693cb8ebe0ae2a94186e4de5b98eca	1	898f75eac5d8bc662604224ebf00c7c9	11/28/2018 16:28
2	806d34dc9ddfafe3a9346a6b268daa1d42693cb8ebe0ae2a94186e4de5b98eca	0	9443ba18ce03fa7d175103fbad05ffa3	11/28/2018 16:27
1	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	e07ef7b72e7bfb20f83b30f2bd3cece1	11/28/2018 16:17
1	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	7b4e321ed7eb4036f625fc6270e1c068	11/28/2018 16:25
1	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	c61cf948447e24857e4e7399446f31eb	11/28/2018 16:26
2	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	e07ef7b72e7bfb20f83b30f2bd3cece1	11/28/2018 16:21
2	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	ea1dbdfe6f5e97d7c80ac3fe739a3678	11/28/2018 16:26
2	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	7b4e321ed7eb4036f625fc6270e1c068	11/28/2018 16:31
2	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	0d6d5677a201917992b1c332b4a42f38	11/28/2018 16:40
2	2b0ab273347390dfc9c04cad43cd2219caed81ed49e0f2691adee615a13210c8	0	f73e7b69205a0b039dfeeebf598153bc	11/28/2018 16:43