ANALYZING LARGE SCALE WI-FI MOBILITY DATA USING SUPERVISED AND UNSUPERVISED LEARNING TECHNIQUES

by

Mohammad Hossein Sarshar

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
December 2016

Table of Contents

List of Tables

List of Figures

Abstract

With the increasing number of Wi-Fi enabled portable devices, and the ubiquitous Wi-Fi networks, analyzing multiple aspects of a population is becoming more insightful, inexpensive and non-intrusive. Network packets propagated from Wi-Fi enabled devices encapsulate spatial, spatiotemporal and behavioral information about the device holders. An opportunity that was available only to online stores a decade ago. In this thesis, we propose two methods to expand the possibilities of Wi-Fi Analytics. First, we present a remote localization technique as an essential preprocessing step to enable Wi-Fi Analytics in the retail and hospitality sector by analyzing non-intrusively collected Wi-Fi packets using supervised learning. Our method is capable of estimating positions without any prior knowledge about the store plan or the antennas' location with only one off-the-shelf access point. Unlike other positioning techniques, instead of estimating a relative position of a device from an antenna, we provide an absolute position for a device as inside or outside of a venue without making any assumption about the site nor the positioned devices. Second, we present a non-intrusive technique to learn about past spatial behaviors of a population by analyzing their SSID data. The main outcome of this component is to expand our knowledge about previously visited locations of a population by collecting few network packets of the Wi-Fi enabled devices and mining the data using unsupervised learning techniques.

List of Abbreviation and Symbols Used

kNN	k-Nearest Neighbors
t-SNE	t-Distributed Stochastic Neighbor Embedding
AOA	Angle Of Arrival
AP	Access Point
API	Application Program Interface
ASCII	American Standard Code for Information Interchange
AUC	Area Under Curve
BSSID	Basic Service Set Identifier
CV	Cross Validation
ESSID	Extended Service Set Identifier
FB	Food And Beverage
GSM	Global System for Mobile
JSON	JavaScript Object Notation
LBS	Location-Based Service
LHS	Left Hand Side
MAC	Media Access Control
NNSS	Nearest Neighbor(s) in Signal Space
OS	Operating System
PC	Principal Components
PCA	Principal Component Analysis

PDF	Probability Distribution Function
POI	Point Of Interest
RFID	Radio-Frequency Identification
RHS	Right Hand Side
ROC AUC	Receiver Operating Characteristic Area Under Curve
RSSI	Received Signal Strength Indicator
SNE	Stochastic Neighbor Embedding
SSID	Service Set Identifier
SVM	Support Vector Machine
TOA	Time Of Arrival
Wi-Fi	Wireless Fidelity
WNIC	Wireless Network Interface Card

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor *Prof. Stan Matwin* for the continuous support of my master's studies and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my master's studies.

Besides my supervisor, I would like to thank my thesis committee: *Prof. Srini Sampalli* and *Prof. Qigang Gao* for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I would also like to thank the Institute for Big Data Analytics, SolutionInc Limited, the Natural Sciences and Engineering Research Council of Canada, Training Program in Big Text Data (CREATE TRIBE), and Big Data for Productivity Congress as the main enablers of this research.

Last but not the least, I deeply thank my lovely wife *Mona* for her patience, supports, motivations, and infinite love, my *Mom* and *Dad* for their endless spiritual supports, kindness, and loveliness, and my *friends* for the joy they complement to my life. I should sincerely and wholeheartedly thank my life-long mentor *Mr. Seyed Hooman Sadat Kiaei* as he supplied me with rich and infinite mental and spiritual guidance in the past 13 years. Finally, I would like to thank my dear and mighty *God* whose purifying, protective, caring, and loving *light* is the source of my strength and perseverance to overcome exhausting moments in my life.

Chapter 1

Introduction

In the big data era, the proliferation of Wi-Fi enabled devices and the considerable amount of data gathered from public Wi-Fi communications have introduced new opportunities in research and industry. Particularly, network packets propagated by portable devices have been very appealing as they encapsulate spatial and spatiotemporal information of the device carrier which can be any moving object like a human or a automobile. Collecting and analysing this type of data which will be explicated in the following chapters, is gaining huge attention as it enables us to collect valuable information from humans across a broad spectrum of industries from retail analytics to public security. The popularity is due to inexpensive, ubiquitous Wi-Fi Access Points (AP) accessible in public venues. They are capable of collecting smart phones' data with little or no modification while serving internet to the public.

Regardless of being connected to a Wi-Fi network, smart phones with an active wireless network interface card (WNIC) repeatedly transmit Wi-Fi packets called probe requests. The packet contains useful information about the device and consequently the device owner, such as the device's Media Access Control (MAC) address, the list of SSID names the phone was previously connected to, the timestamp of the message propagation and the received signal strength indicator (RSSI) to name a few.

Portable Wi-Fi enabled computing devices are becoming more popular in the entire world. This number is significantly large for developed and developing countries. In a report by Cisco [22], there were close to 3 billion smart phones and 97 million Wi-Fi enabled wearable devices in 2015, globally. Cisco believes that there were 64.2 million public Wi-Fi hotspots in 2015 that this figure is predicted to grow approximately 600% to 423.2 million by 2020. Based on this report, users also prefer Wi-Fi over other alternatives as 51% of the entire network traffic was through Wi-Fi networks in 2015 which is predicted to reach 53% by 2020.

The possibility of collecting probe requests from this large population of Wi-Fi

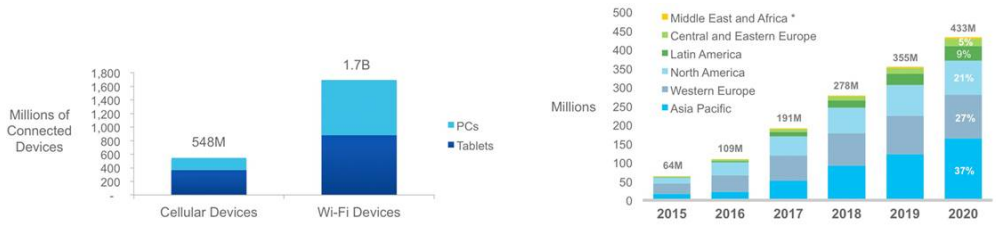

Figure 1.1: The left figure shows the number of devices (PC and Tablet only) that will be connected to Wi-Fi networks by 2020. The plot on right shows the growth of public Wi-Fi by region from 2015 to 2020. (Source [22])

enabled portable devices, unveils interesting information about the device holders. For example [29, 30, 57] show that the spatial data can accurately unveil complex characteristics of individuals and groups. Moreover, there have been studies on analyzing coarse-grained and fine-grained spatial and behavioural activities of the customers known as Physical Analytics [64], such as obtaining customers relationships [18] measuring the engagement index of customers [56], predicting dwell time [49], quantifying campaign performance and forecasting the number of customers and pick hours. Therefore, mining the raw data gathered from smart phones provides venue owners with different layers of visibility on the performance of their stores. This motivates several tech companies to leverage this massive opportunity to provide actionable insights to venue owners. For instance, Euclid Analytics [1], AisleLabs [2], TurnStyle [3], and Purple [4] have built successful platforms known as Wi-Fi Analytics to provide Wi-Fi data rich insights to brick-and-mortar businesses. Many providers of turn-key Wi-Fi solutions are also interested in building tools that will supply their customers with knowledge about their clients' behaviors. This thesis is a report from one such project between the Institute for Big Data Analytics and the SolutionInc Limited.

In this thesis we demonstrated two prominent aspect of Wi-Fi Analytics Platforms that can extend the potentials of such systems. As first, using RSSI value of the probe requests we propose a novel positioning system to identify the location of the users. Second, by mining another field in probe request known as SSID, we explore the spatial behavioral similarities of a population.

Wi-Fi Analytics is a double-edged sword. In spite of massive opportunities that analysing Wi-Fi data creates, it can be a hazardous technology. One of the most

controversial outcomes of analysing Wi-Fi data, particularly public Wi-Fi data, is the possibility of privacy breach. Although this aspect is not included in the scope of this thesis, it is very important to mention that preserving privacy of individuals is of the utmost importance. In a collaboration with the *Institut de Recherche en Informatique et Systmes Alatoires* and the *Institut National des Sciences Appliques de Lyon* we are working on several techniques[8] focused on Wi-Fi data to preserve the privacy in Wi-Fi Analytics tasks.

1.1 Aim and Objectives

Based on the provided introduction, the main aim of this thesis is to extend the possibilities of Wi-Fi Analytics platforms by introducing: 1) *a non-intrusive absolute indoor localization technique system* and 2) *a non-intrusive framework to identify population spatial behavioral characteristics from partial spatial historical data.*

In order to address aim, the following objectives are identified:

1. Investigate and survey the existing methods proposed for indoor localization systems.
2. Propose a new method for indoor positioning system to be used in brick-and-mortar domain.
3. Investigate and survey existing works on analyzing SSID.
4. Propose a framework to capture the spatial behavioral characteristics of a population from SSID data.
5. Demonstrate the results and effectiveness of the proposed methods.

1.2 Contributions

The contribution of this thesis is twofold. The first contribution of the thesis is based on the previously published work of the thesis author [61]. The theoretical and empirical aspects of the published work has been updated and provided in this thesis. The second contribution is based on an unpublished research project as an extension to the theme of the thesis. The contributions of this thesis is provided below:

- Investigation of two sources of data to be used as the data sources of indoor positioning system.
- Proposal of the framework of an indoor positioning system based on the investigated data sources, a novel framework as an essential part in the *prepossessing* stage of Wi-Fi Analytics platforms.
- Implementation of the proposed indoor positioning system.
- Evaluation of the proposed indoor positioning system using visualization and quantitative approaches.
- Investigation of methods to map SSID data to actual locations, a method that yields *partial spatial historical*.
- Proposal of a framework to cluster a population based on their *spatial behavioural features* extracted from the partial spatial historical data, a novel framework that has application in various domains.
- Implementation of the proposed clustering framework.
- Evaluation and illustration of the results of the proposed clustering framework using visualization and quantitative approaches.

1.3 Thesis Outline

The remainder of the dissertation is organized as follows. In Chapter 2 we provide a review on the background of positioning system and its types. Then, we provide the related works in this field. In the second part of Chapter 2, we explain the background and the past research works that have been published on SSID data, in general.

In Chapter 3 we present our positioning system and the proposed crowdsourcing technique. First, we explain the problem statement and an overview of the proposed positioning system in Section 3.1. Then, we explain the main method in Section 3.2. We define our crowdsourcing datasources in this section in addition to the noise cancellation (Section 3.2.1), feature engineering (Section 5.3.1), and hypothesis validation methods (Section 3.3). And finally, in Section 3.4 we explain our positioning system based on the crowdsourced data.

In Chapter 4, we provide the evaluation results of the positioning system. First we investigate its performance in position estimation task in Section 4.1. Then, we provide the performance of the positioning system in transfer learning task in Section 4.3.

In Chapter 5, we present our method to cluster a population using SSID data. First, we provide the problem statement and an overview of the proposed solution in Section 5.1. Then, we provide our method to map SSID data to its actual location in Section 5.2. The main method including feature engineering (Section 5.3.1), feature vectorization (Section 5.3.2), feature normalization (Section 5.3.3) and the clustering task (Section 5.3.4) is provided in Section 5.3.

In Chapter 6, we provide the evaluation results of the clustering task. First, in Section 6.1, we present our evaluation methods that we employed to evaluate the clustering task. In this section we explain the Silhouette Analysis technique. Then, in Section 6.2, we present the employed data visualization technique. Finally, in Section 6.3, we introduce the final result of the clustering task and provide a deep explanation of the result.

Lastly, in Chapter 7, we present the conclusion for both the positioning system and the clustering task on Partial Spatial History in addition to the possible future improvements of the proposed methods (Section 7.1.1 and 7.2.1).

Chapter 2

background and Related Work

In this chapter we provide a survey on two different problems this thesis wants to solve. First, we investigate the related research work in the field of indoor positioning systems. Then, we provide a survey on analysing SSID data. The former is a well studied research field while the latter has a shallow depth as it could attract the attentions of a few researchers.

2.1 Positioning Systems

Due to the wide application of positioning systems, this field has attracted many attentions in the pas two decades. The application of indoor positioning system appears when tracking a moving object is of great interest such as tracking automobiles, planes, humans, ships, drones, etc. To solve many of the positioning challenges, Global Positioning System (GPS) is a great answer. However, there are several scenarios in which this service is not practical such as tracking a moving object in indoors. This is due to the inaccessibility of GPS signals in indoor settings [73]. As a result, conventional positioning systems are not appropriate for such situations. Due to the huge application of indoor positioning systems in a wide range of problems, this field has attracted many attentions in research and industry.

The main objective of a positioning system is to estimate the positioning of a moving object in a 2D or a 3D surface. In other words, the **tracker** is responsible to estimate the position of a **trackee**. For example, a GPS enabled device is capable of estimating its position from the received GPS signals. To achieve this objective, a positioning system requires two hardware devices: a signal transmitter and a measurement unit. The transmitter might be a GPS satellite that transmits GPS signals and a measuring unit can be a GPS enabled device that measures its global location from the characteristics of the received GPS signals. In indoor setting, in spite of the maturity of this research field, there is no universally accepted choice for these two

components. The main reason is that each choice provides a unique level of cost and accuracy which is completely domain dependant.

The diverse solutions can be grouped into four different topologies[28]. In the first topology, the trackee and the tracker are one device and which is the signal receiver. The signal transmitter, such as a GSM tower, does not have any responsibility to estimate the position of the signal receiver. This topology is called *self-positioning system*. In the second topology, the transmitter is the mobile device and the measuring unit is one or several fixed devices that estimate the position by measuring the received signals. This topology is called *remote positioning system* as the signal receiver is the *tracker* and the signal transmitter is the *trackee*. The third topology is called *indirect remote positioning system* if a self-positioning system sends its position data to a remote location. Finally, the fourth topology is called *indirect self positioning* if the remote positioning system sends the position data to the trackee.

Two very famous indoor self-positioning systems are *indoors* and *iBeacon*. *indoors*[1] is a dedicated positioning and navigation solution that operates as a self-positioning system and is capable of positioning and navigating a smart phone user in a roofed environment. *iBeacon*[2] is an Apple[3] product which is built to estimate the distance of a smart phone from the iBeacon. As a self-positioning system it is used to estimate the position of a device from an iBeacon in a roofed environment.

In Wi-Fi Analytics domain, where position of devices as the indicator of customers' positions is of great interest, a low cost and reliable method is required to achieve this data. The first topology requires the analytics platform to access the device to obtain the positioning information. However, generally, Wi-Fi Analytics platforms should be able to obtain this information independently. The third topology, needs the device to calculate the position using its inertial sensors and contribute its position to the analytics platform. This topology is not practical because in this case, the platform becomes completely dependant upon the wish of the device owner as the device should willingly contribute its location to the measuring unit. However, the second topology can work independently as the measurement can occur remotely at the tracker unit while the trackee should only transmit signal than the position information. The

[1]https://indoo.rs/solution/navigation/
[2]http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/
[3]http://www.apple.com/

fourth topology is not of interest as the location does not need to be passed to the trackee in this domain. Therefore, the second topology suits best with the domain of this thesis.

2.1.1 Signal Features for Indoor Positioning Systems

There are a number of features that can be extracted from a travelling signal to help estimating the positioning of the signal transmitter. These features can be grouped into three main categories:

1. **Power:** This type feature is a physical feature obtained and measured at the receiver which indicates the power (not the quality) of the received signal. One of the popular power features is RSSI which is accessible in a wide range of wireless mediums such as Wi-Fi, Bluetooth [52], and ZigBee[31]. The power feature is measured at the signal receiver for each transmitted message. However, this feature is unreliable due to multipath effects. Multipath effect is referred to the phenomenon in which the travelling signals pursue different paths to the receiver at each packet transmission due to the complexity of the indoor environment. This happens particularly in situations where there is no Line-Of-Sight (LOS) that signal may collide with several surfaces before reaching the signal receiver. This makes a single power feature unreliable in estimating the position. Similar to the proposed method in this thesis, fingerprinting techniques, can offset this drawback and obtain reliable results.

2. **Time:** The Time Of Arrival (TOA) and Time Difference Of Arrival (TDOA) are popular time features. In LOS conditions, TOA is proportional to the distance between the transmitter and the measuring unit. In order to measure the accurate distance between the units, the system time in both components should be synchronized and the transmission time should be included into the transmitted packet. The TDOA is to measure the relative distance of a transmitter by measuring the time difference of the signal arrival $w.r.t$ to the TOA of signals of other transmitters. Therefore, by TDOA feature, the measuring unit can estimate the distance of the transmitters relative to each other. These features, rely heavily on LOS conditions as in complex indoor settings, they

cannot provide reliable values.

3. **Angle:** The angle feature is the angle of arriving signal which combined with the distance, can provide the position data of the transmitter. Angle of Arival (AOA) is a popular angle based feature. Calculating angle based features requires directional antennas or Multipath Input Multipath Output (MIMO) interface (such as antenna arrays) on the measuring unit. The cost of dedicated infrastructure makes it a costly and impractical in some scenarios.

2.1.2 Position Estimation Techniques

In a remote positioning system, the measuring unit uses several techniques to estimate the position of the trackee. These techniques try to estimate the position by utilizing the features extracted from the flying signal in two way. We divide the estimation techniques into two main categories: *Geometric Mapping* and *Fingerprinting* position estimation techniques.

1. **Geometric Mapping:** This method builds geometric models to calculate the location of a device from geometrical features of the propagated signal. Using *triangulation* amongst multiple reference points (three and more), this method estimates the position of signal transmitter from the extracted features: time, power or angle. A trivial implementation of this method is to find the intersection point of the circles of TOA. In other words, assuming TOA feature draws a circle around a reference point e.g. an antenna, having three reference points which have three circles, the intersection of the circles indicates the approximate location of the signal transmitter (Figure 2.1). The combination of TOA and AOA makes geometric mapping technique more efficient. For example, calculating the distance at the measuring unit from TOA and the angle of arrival provides a line that points to the signal transmitter (Figure 2.3). It is believed that this model work well in LOS conditions while considerably suffers from complex indoor environments where LOS is not available [38, 45].

2. **Fingerprinting:** Another approach in indoor localization is called *fingerprinting* or *scene analysis* [47] in that a database from the site is surveyed to populate a database of known locations with the corresponding radio signal

features. The site survey task is known as the offline stage and is an essential part of this method. Then from the populated knowledge base, new instances are classified known. The classification of the new instances is called the online phase. This model uses statistical or machine learning techniques to position new unseen observations. The offline phase of this method requires full access to the site as it is required to survey multiple instances of signal features e.g. RSSI from each position. However, this method is very costly as it takes a considerable amount of resources and time to collect adequate number of data points for each spot of a location to obtain a reasonable classification results. Although the fingerprinting approach provides reasonable accuracy, the offline phase makes it unattractive. To overcome the offline/site survey phase, a method called *Crowdsourcing* is introduced in which it relies on the public to contribute their signals and labels to populate the database. For instance, a positioning system may require a number of users to install a site survey app [58] that reports the location of the devices to a measuring unit. Collecting a large number of crowdsourced position information reduces the efforts of offline phase.

Because of the popularly of Wi-Fi networks, and the proliferation of the Wi-Fi enabled smart devices, Wi-Fi analytics platforms leverage this opportunity to track customers by tracking the smart devices. Tracking the position of the devices is an important step to provide deep and accurate insights about the customers of brick-and-mortar businesses. As a result, Wi-Fi analytics platforms use APs to measure the position of Wi-Fi enabled devices. Similarly, the positioning system proposed in this thesis, is a remote positioning system that estimates the Wi-Fi enabled devices using fingerprinting approach by measuring the RSSI value of the packets transmitted by trackee and received at AP level as the tracker (the measuring unit). Moreover, the offline stage of our proposed method is equipped with a crowdsourced technique which is explicated in Chapter 3. Therefore, in the rest of this section, we focus on related research works that are implemented based on RSSI feature, employ fingerprinting and utilize crowdsourcing for their proposed methods but first we explain the theory behind the RSSI value.

RSSI value is in a range starting from 0 to -120. RSSI values closer to -120 are

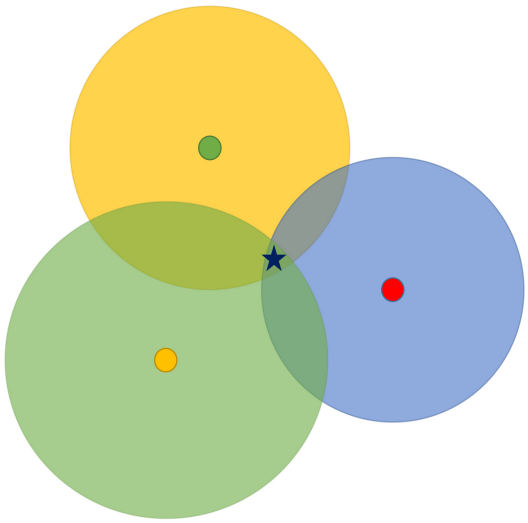

Figure 2.1: An overview of a triangulation method using three reference points (circles) e.g. three antennas to estimate the position of the transmitter (start) e.g. a smart phone. The larger circles surrounding the smaller circles are the distance features such as TOA. The intersection of the three circles is the approximated position of the transmitter device.

considered stronger. This feature is calculated from the voltage of the signal received at the receiver unit using the following equation:

$$V = \sum_{i=1}^{N} \|V_i\| e^{-j\theta_i} \tag{2.1}$$

where V is the signal voltage of all the signals received at the measuring component

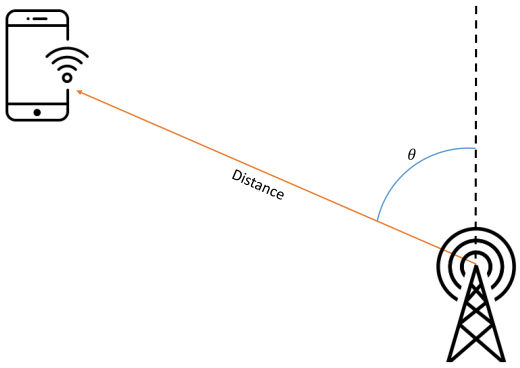

Figure 2.2: An overview of a the combination of angle and distance features to locate the transmitter device. The θ is the angle of the received signal (AOA) and the orange line is the distance which is measured by time feature (e.g. TOA).

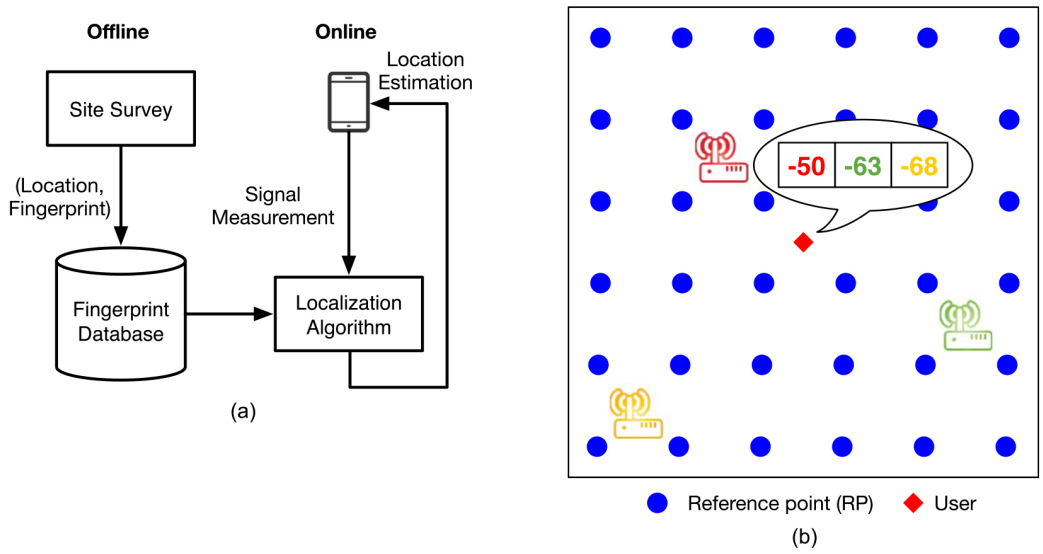

Figure 2.3: Figure (a) shows an overview of a positioning system. It first collects the site survey to populate a database of fingerprints. Then from the database, using a localization algorithm (e.g. classifier), it estimates the position of an unseen device. Figure (b) shows the positioning system in action that the blue points are the surveyed positions, measuring unites are provided by icons of AP and the red dot indicates a new unseen device with specific RSSI values from each measuring unit. Then a localization algorithm will identify the location of the device based on the measured values and the populated database. (Source [35])

once a signal is received. Due to the multipath effect, the measuring unit receives N signals from a device for a single packet. V_i indicates the voltage of the i^{th} multipath component of the original signal and N is total number of components. RSSI value is calculated from the computed voltage value using the following equation in decibels (dB):

$$RSSI = 10\log_2(\|V\|^2) \tag{2.2}$$

Most of the Wi-Fi based fingerprinting positioning systems use RSSI value to estimate the position of the signal transmitter [9, 67, 41, 11, 74, 46, 21]. Padmanabhan et al. proposed Radar [9, 67] which is known to be the first introduced positioning system based on RSSI values. Their research proved that fingerprinting approach offsets the unreliability of RSSI values and makes it a reliable feature for indoor positioning systems. They introduced the site survey in offline phase to populate the Radio Map of a location and employed Nearest Neighbor(s) in Signal Space (NNSS) in

the online phase to estimate the position of a new unseen device. Their investigation shows that their proposed method performs well in complex situations with 5-10 meter error. This work is the main reference point for the other fingerprinting positioning techniques that were proposed afterwards.

Saxena et al. [25], proposed a positioning system based on the RSSI value of the received signal at the AP level. As a fingerprinting positioning system, the RSSI fingerprints were collected from several points of a site to generate a grid of known locations. Then, the mean of the RSSI fingerprints for each reference point is calculated as the feature for the positioning system. Using kNN algorithm, they classify the new unseen observations based on the RSSI for the new device. Their analysis shows that this method is capable of estimating the positioning of a new unseen device with 1.1 meter accuracy 90% of the time.

Chen et al. [17] presented a Wi-Fi based fingerprint method for Location-Based Services (LBS). They introduced two different methods in their paper. First, they proposed a fingerprinting approach based on the RSSI value received at the AP to train a kNN classifier as the positioning system and second, they proposed a distance-based trilateration method using three APs to detect a new unseen device. By building a radio map of RSSI values for several points of their test location, they compared the performance of both methods. The results showed that fingerprinting approach outperforms the distance-based trilateration approach and the combination of both can achieve better results.

Al-Ahmadi et al. [7] proposed a Wi-Fi based fingerprinting positioning system in multi-floor situation. They first investigated the RSSI value characteristics such as the statistical distribution and its unreliability in multipath effect. Then they studied the RSSI values in multi-floor situation to justify their method accordingly. In the offline phase, they collected samples from multiple floors of a university. Using Bayesian graphical model, they trained their model using the collected fingerprints. The model is tested using four sets of Markov chain Monte Carlo sampling techniques. Their investigation shows that their method is capable of positioning a device with 2.3 meter accuracy in multi-floor environments.

Khodayari et al. [39] proposed a RSSI fingerprinting approach known as predicted K-nearest-neighbor (PkNN). As its name suggests, they use kNN in the online phase

as the trained classifier to predict the unseen devices. In order to estimate the position of a device, they take into account the K found neighbours, the past visited location and the speed of the device. The evaluation of the PkNN method shows that it outperforms the kNN method by 33% with 1.3 meter improved accuracy.

In [32], Fang et al. proposed a fingerprinting positioning system based on Principle Component Analysis (PCA) in which the RSSI values are transfomed into principal components (PC) to efficiently utilize the information of each AP. This method considers the PCs instead of APs for the positioning system. The investigation of this approach shows significant improvement in accuracy and computation time in online phase.

In order to avoid the expensive task of site survey, a new method called *Crowdsourcing* is introduced. For example, Bolliger [13] proposed the very early crowdsourced methods called *Redpin* which builds fingerprints exclusively from user collaboration. Redpin assumes that people install location information reporting system on their phones and report their RSSI fingerprints to the positioning system. Bhasker et al. [12] proposed a method that requires user interaction on a map to provide their geolocation information to a positioning system. Then from the provided information, the positioning system builds a model to predict new unseen devices. Lee et al. [44] proposed a similar crowdsourced method that requires a software be installed on the smart phones of the crowds and requires their consent to contribute their positioning information to the positioning system in addition to collecting the RSSI fingerprints. Then, from the contributed fingerprints, it trains a kNN classifier to predict the new unseen devices.

Wu et al. [71] proposed an advanced crowdsourcing method called Locating in Fingerprint Space *LiFS* in which it transforms the site plan into a stress-free floor plan for location estimation task using graph theory. Then, it maps the already collected and unallocated RSSI values to the floor plan based on a proposed mapping process. Then using kNN algorithm, it estimates the position of a new unseen device. The experiments show that LiFS has low cost of implementation, with maximum of 6 meters of error.

Our proposed indoor positioning systems, as an enabler of Wi-Fi Analytics, tries to answer a very fundamental question about users' absolute position. Using Wi-Fi

signals, we would like to identify users' location as inside or outside a store with no knowledge about the floor plan nor the position of the antenna and with only a single off-the-shelf AP. In addition, we would like to replace the site survey in offline mode with a method which is not only crowdsourced but non-intrusive. Previous works that are based on fingerprinting approach are able to answer the absolute question. In addition, the proposed fingerprinting methods also use a machine learning classifier to estimate the position of a new unseen device. In these regards our work is similar to [9, 67, 41, 11, 74, 46, 21, 25, 17, 7, 39, 32]. However, the main difference between our method with the provided works is that they require a heavy site survey phase as we call assumption about the site which is not required in our method. However, the provided research works have an advantage over our proposed method which is the relative (fine-grained) position of a device which is not achievable in our technique.

The previous crowdsourced techniques also require heavy user interactions or their collaboration to contribute their position information to the positioning system [58, 13, 12, 44]. Our proposed method has an advantage in this regard as it collects the crowdsourced position data non-intrusively which makes it different from the provided research works. Another provided crowdsourced method [71] does not required user interaction but requires the site plan. In this regard, our method is superior as we do not require any information about the site plan nor the location of the antenna. However, this method has one advantage over our method as it requires no labeled data for the positioning system.

Finally, our system is capable of operating using one off-the-shelve AP and does not require dedicated devices for the position estimation task which makes it dissimilar to the *iBeacon* and *indoors* solutions.

2.2 Mining SSID Data

Due to the popularity of the Wi-Fi Analytics to provide business insights from Wi-Fi data has attracted many of the attentions in industry and academia. Analysing SSID data as one of the extensions of Wi-Fi data has been appealing to both sectors in the past decade. In this section, we investigate the previous attempts that leverage SSID data to provide insights into a population. First, we start with explaining SSID data.

Whenever a Wi-Fi enabled device makes a connection to a Wi-Fi network, it

saves the name of the Wi-Fi network known as *Service Set Identifier* (SSID, *a.k.a* ESSID) in a list called *Preferred Network List* (PNL). PNL contains the SSID names of all the previously connected networks. In case a device moves from one location to another, it searches the available networks by sending a packet called *probe request*. Inside the packet, the device includes several information about itself including MAC address, device type, and OS. In addition, it includes one of the records from the PNL list inside the probe request which is known as SSID field. If the packet is received by the AP which is hosting a Wi-Fi network with similar name to the transmitted SSID name, it sends a packet called probe response to indicate its presence. Then the device starts the connection establishment if it decides to. Transmitting probe request is not limited to the connection establishment process. An active WNIC repeatedly and blindly sends the probe request into space. Therefore, by collecting several probe request packets from a device, we can capture a number of SSIDs from its PNL list.

Before providing the literature review we should mention one point. In general, the literature of the works on the SSID data consists of into two aspects: 1) highlighting the concerns over privacy and the threats of SSID data and 2) application of SSID in different fields. Most of the search works, tried to cover both sides.

One of the most popular and viral examples of practical usage of SSID data is the Snoopy project [69, 70]. It successfully highlighted the opportunities and the threats SSID data can provide and cause. This project captured many attentions as several conferences, video tutorials, and blogs that covered it went considerably viral[4]. In this project, Glenn Wilkinson built a framework to show the insights SSID data can provide from different angles of view like security, privacy, and business. He programmed several drones hovering a crowded neighborhood in London, UK to capture SSID data from the probe requests of the crowds. Sending captured data to a back-end server, it can identify the past visited locations of the crowd. He argued that this has many applications in public security and business. Similar to Wilkinson's work, Bonné et al. [14] built a platform named SASQUATCH to study the privacy threats of the SSID data and to increase the public awareness for such threats. By displaying the live results of the platform showing the past visited locations of the

[4]The DEF CON 22 coverage at YouTube → https://www.youtube.com/watch?v=knrvrR-B1ZI

people passing the display by mapping their SSID data to actual locations, they presented the level of information they managed to achieve from the public in a non-intrusive manner. The displayed information shocked many of the passersby as 90% expressed their fear that others can view their past visited locations, 60% of the participants signed up to receive updates to help improve their privacy and 83% where not aware of such threat.

Inspired by Snoopy project [70], Chernyshev et al. [19] investigated the hypothesis that whether in aggregate view SSID names can unveil personal or professional information about the device owner. They validated their hypothesis by first mapping the SSID names to actual locations using Wigle and Google Places API (more detail is provided in Section 5.2.1). Then, by Named Entity Recognition (NER) techniques they tried to discover semantic information from the SSID data. Then, they studied the quality of the mapping process. The result of their investigation shows that 49% of the SSIDs are identifiable and potentially provide some information about the owner of the device such as past visited locations or even names. Although they discussed that SSID data is not very reliable and accurate, they believe that it provides a level of inference for identification of linkage between devices and their owners.

Chunche et al. [24] proposed a framework to find social links amongst people from their SSID data. Their research is based on a hypothesis that people with similar and low frequent SSIDs should have strong social links. They investigated several methods to validated their hypothesis. Amongst the employed similarity metrics, cosine-IDF achieved the highest score to identify the social links. The cosine-IDF similar to TF-IDF technique offsets the effect of highly frequent SSID names in the entire SSID dataset. In addition, they discussed the application of their proposed method for advertisement and forensic purposes. Barbara et al. [10] proposed a similar research work to infer social and socioeconomic status of a population through their SSID inspired by a graph-based model known as affiliation network [42]. They discovered that the built social graph has similar structure to well-know social structures. Other contribution of their research work is to infer the language of the population and the social class.

Seneviratne et al. [62] presented a research work to find SSID links to business entities using word similarity metrics. First, they discovered the names of businesses

in the neighbourhood of data collection center. Then using word techniques such as *TF-IDF* and *cosine* similarity they tried to discover the actual location of the SSID. Their result shows that word level cosine similarity has the highest score to link SSID to its actual business venue with 97% precision. Similarly, Di Luzio et al. [48] proposed a research work to deanonymize the location such as province, city and address of a location from the SSID with a simple search in Wigle dataset. Then by comparing the results with their ground truth dataset they showed high possibility to deanonymize the actual locations from the SSIDs of a population.

Similar to [10], we would like to find the social structure of a population through mining their SSID data. However, instead of direct utilization of SSID names, we considered the location types to find the semantic similarities of a population. In addition, we used several information theory and unsupervised learning techniques to prepare, process and visualize the data that has not been done before. In order to find the location type, we mapped the SSIDs to their actual locations using an online location search API that in this regards our work is similar to all the provided research works.

Chapter 3

Absolute Wi-Fi Positioning System

In this chapter, we aim to explain our proposed localization system which is capable of classifying people as inside vs. outside $w.r.t$ a location. First, we explain the problem statement and its importance. We continue by explaining the nature of datasets and the necessary preprocessing steps before proceeding to the main method. Then we explain our method for the localization technique.

3.1 Problem Statement

A probe request is a packet which is sent by any Wi-Fi enabled device with an active WNIC. An antenna, such as an AP, can capture those probe requests. If the antenna is located in a crowded location, like a store, it can capture several packets from the devices in its proximity. Probe request contains several information about the source device such as MAC address, operating system (OS), the device model, RSSI [1], etc. This data is interesting to the venue owner as it provides some quantitative information about the customers in a non-intrusive fashion by feeding this data into a Wi-Fi Analytics platform. A typical task of such platforms is to count the number of unique MAC addresses to estimate the number of customers in a given time. However, a naive implementation of this trivial task can return a non-realistic number which is much higher (2-20 times) than the actual number of customers. The reason is that the captured probe requests are originated by devices located at the proximity to the antenna. It can be from a car which is parked outside the store or a customer in another store a block away from the capturing site. This problem is more significant in busy neighborhoods. Therefore, the platform must be capable of identifying the location of the observed devices as whether they were inside or outside of a location to provide more accurate insights. We call such capability, absolute localization system

[1] Although we discussed in Section 2.1 that the RSSI value is calculated at AP level for each receiving packet, in the rest of the this thesis, we assume that the RSSI value is included inside the probe request packet to avoid confusion.

as it identifies the location of a device as inside or outside $w.r.t$ a location instead of a relative location of the device from the antenna. We achieve this goal by leveraging the characteristics of RSSI to identify the location of a device.

We define the objective of our proposed positioning system as: let A be the list of locations where we capture probe requests and $\alpha \in A$ be one of those locations, U be the set of all captured users[2] and $u \in U$ be a unique user, $\phi_{u,\alpha}$ is a set that contains the entire captured probe requests of user u during a visit to location α is denoted by $\phi_{u,\alpha} = \{O_{t_1}, O_{t_2}, ..., O_{t_p}\}$ where data point O_{t_j} is the RSSI of a probe request received by the antenna at location α at time t_j for user u, we would like to classify the visit $\phi_{u,\alpha}$ of user u as inside or outside of the location α. An overview of the process is illustrated in Figure 3.1.

In other words, the initial challenge before applying any analytics on the raw data is to identify the signals amongst the noise. APs and antennas are capable of capturing probe request packets in proximity where it can be from a passerby in the street across from the store or a real customer ordering a take away food at the store. From a real world dataset we found that in some cases 85% of the captured data were from outside users. Considering this, an indoor localization step is essential at the preprocessing stage to label the observations as outside people or inside customers before being able to extract any useful information from the data.

Indoor localization is a well studied research field in which its main objective is to gain a level of accuracy based on the problem domain. The accuracy can be from room- to decimeter-level [66] using signals available in indoor settings (e.g. Wi-Fi, GSM, Bluetooth, and RFID) or smart phone built-in sensors (e.g. accelerometer, and gyroscope). However, these methods suffer from several limitations and assumptions that make them impractical in this domain. Existing solutions assume that they have either full access to the users' devices to read smart phones' inertial sensor data, or have full access to the venues and enough resources to collect fingerprints from every corner of the store. In some methods they need to equip the location with multiple access points, or cameras. We believe that in many real-world cases, similar to the aforementioned companies, these methods are not practical. Having a large number of clients, a generic method with low computational and technical complexity is required

[2]From here by using the term **user** we mean the Wi-Fi enabled device which is carried by a person unless otherwise stated

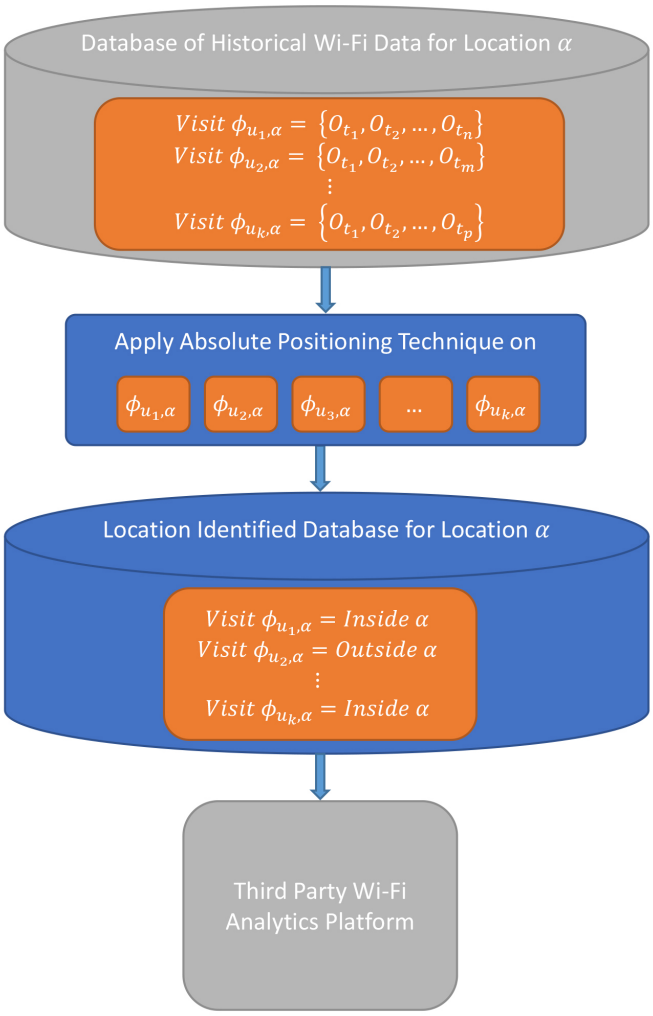

Figure 3.1: The framework for the absolute positioning system to classify the entire visits of users $U=\{u_1, u_2, ..., u_k\}$ at location α as inside or outside. The main process, which is painted in dark blue, classifies each visit as inside or outside based on the given historical dataset. Finally, the a new dataset is retrieved which contains the position information of each visit. This dataset can be an input to any Wi-Fi Analytics platform for further analytical tasks or can be used to identify the location of an unseen user.

which is applicable in different settings.

In this chapter we introduce a generic method proper for the essential preprocessing stage of other Wi-Fi Analytics tasks. This non intrusive method assumes that the localization system has no access to the venue nor the device; the whole process happens remotely at AP level. Moreover, it requires a single AP to manage the localization task. The test results showed high accuracy and easy adaptability. We tested

the method in different scenarios using real world Wi-Fi datasets gathered from 5 different restaurants and cafés . By means of this method, Wi-Fi Analytics platforms are able to diagnose the coarse grained location of customers with the maximum flexibility and adaptability.

3.2 Method

In this section, we explain the method for the aforementioned problem. This positioning system is a fingerprinting approach (Section 2.1) where data is provided by crowdsourcing. A fingerprinting positioning system relies on the dataset of RSSI values collected from different spots of a location. Each location is considered to be a label or class for the captured dataset. By building a classifier using this labeled dataset, a positioning system becomes capable of classifying the location of a Wi-Fi enabled device. As a Wi-Fi platform, this method is very costly especially considering having several customers there should be several locations to be scanned. We investigated two sources of data as crowdsourced datasources to avoid this expensive data collection process.

Most of the public Wi-Fi networks in brick-and-mortar businesses require a simple registration process before granting access to its network users. This process generates a dataset known as *Authentication* dataset that contains the MAC address of the registered device and the time of the registration. Considering the Wi-Fi RSSI values of users who registered at Authentication dataset, we can assume these devices as the inside population, therefore, we have the labeled data for the inside class. In addition, most of these locations, are closed at night. Hence, if any packet is received at that period, we assume that it should be originated from the outside population. Therefore, we can generate the labeled data for the outside population. These two classes can build a complete map of inside and outside population to build a comprehensive positioning classifier. Our indoor positioning system is based on these two hypotheses, that in this chapter, we try to validate. Firs, let's describe the datasources:

Wi-Fi Dataset: This datasource contains the probe requests collected from 5 restaurants and cafés located in Halifax, Canada. The dataset consists of 43,216,265 probe requests for 472,497 unique devices in the past three years. For reasons of confidentiality, these locations are labeled as location A, B, C, D, and E. Data was

gathered from users in the proximity to APs installed at those locations. As explained above, the collected packets can be originated from inside and outside devices. Each location is equipped by one off-the-shelf AP[3], that its in-store position is unknown to us. The APs operated continuously 24/7 for the whole period and recorded the entire received probe requests. The RSSI value for each probe request is calculated at the AP in the arrival time. In average, each device propagated a packet in every 3 minutes. The dataset was collected completely non-intrusively with no user interaction; it only needs a person to hold a Wi-Fi enabled device with an active WNIC.

Definition 1 *(Wi-Fi Dataset) Let* $W_\alpha = \{w_{\alpha,1}, w_{\alpha,2}, ..., w_{\alpha,n}\}$ *be the Wi-Fi dataset for location* α *containing* n *data points where each data point* $w_{\alpha,i}$ *representing a probe request that is a 3-element tuple denoted by* $w_{\alpha,i} =< MAC, RSSI, Time >$.

Dwell Period: W_α dataset consists of several observations denoted by $w_{\alpha,i}$ which contains the MAC address, the RSSI and the time of the observation. From this dataset, another dataset called *Dwell Period* is generated to represent each visit of user u.

Definition 2 *(Dwell Period) Let* $\alpha \in A$ *be a location and user* $u \in U$ *be a user,* $\phi_{\alpha,u,i} = \{O_{t_1}, O_{t_2}, \ldots, O_{t_p}\}$ *is the* i^{th} *Dwell Period of user* u *in location* α *where data point* O_{t_j} *corresponds to the RSSI value of the probe request captured at time* t_j *from device* u *at location* α *and* $\forall t$ $(t_1 < t_2)$.

Dwell Period is an ordered set which contains the probe requests of user u at location α. In other words, $\phi_{\alpha,u,i}$ contains the RSSI value of the probe requests captured from user u for her i$^{\text{th}}$ visit to location α. The AP at location α should receive at least one signal from a user u within a window of 10 minutes to be considered a continuous Dwell Period and place the probe request in the i^{th} *Dwell Period*. Hearing back from the device u after 10 minutes of silence is considered the beginning of a new $\phi_{\alpha,u,\mathbf{i+1}}$ dataset. As a result, we may have several Dwell Periods for user u and location α if user u where at the proximity to the antenna at location α for several times. The combination all *Dwell Period* for location α generates a larger dataset denoted by $\Phi_\alpha = \{\phi_1, \phi_2, ..., \phi_n\}$ where $\phi_i = \phi_{\alpha,u,i}$ is a shorter symbol to name a Dwell Period.

[3]The brand of APs for all 5 locations is Meraki MR12

Definition 3 *(Dwell Dataset) Dwell Dataset is denoted by* $\Phi_\alpha = \{\phi_1, \phi_2, ..., \phi_n\}$ *where data point* $\phi_i = \phi_{\alpha,u,i}$ *is a Dwell Period.*

Authentication Dataset: Another source of data called *Authentication Dataset* is populated at the moment when user u logs in to the public Wi-Fi network of location α. This dataset contains device MAC address, and the time of login.

Definition 4 *(Authentication Dataset) Let* $K_\alpha = \{\kappa_1, \kappa_2, ..., \kappa_n\}$ *be the Authentication Dataset where data point* $\kappa = (m_u, t_j)$ *contains MAC address* m *of user* u *who logged in to the network at time* t_j*.*

Authentication Dataset contains the information of users who logged in to the public Wi-Fi network of location α. It is possible that user u has multiple records in dataset K_α if she logged in to the network multiple times. Similar to *Wi-Fi Dataset*, the *Authentication Dataset* is populated non-intrusively but with a minimal user interaction. In this crowdsourcing approach, data is gathered in a normal process of users connecting to a public Wi-Fi network, with no intention of contributing their spatial data to a localization system.

Registered Data: The combination of *Dwell Dataset* (Φ_α) and *Authentication Dataset* (K_α) for location α can generate another dataset as we call *Registered Dataset* and denoted by Π_α where $\Pi_\alpha \subset \Phi_\alpha$. Let the time of the first and the last probe requests O_{t_1} and O_{t_p} where both $\in \phi_i \in \Phi_\alpha$ be t_1 and t_p, respectively, the MAC address of ϕ_i be m_i, the time of an element $\pi_j \in \Pi_\alpha$ be t_j, and the MAC address of π_j be m_j, then ϕ_i is a member of Registered Data Π_α if the following conditions satisfies:

1. if $t_1 \leq t_j \leq t_p$
2. if $m_j = m_i$

Definition 5 *(Registered Dataset) Registered Data* Π_α *for location* α *is denoted by* $\Pi_\alpha = \{\pi_1, \pi_2, ..., \pi_n\}$ *where* $\Pi_\alpha \subset \Phi_\alpha$ *and* $\pi_1 \in \Phi_\alpha$*.*

In words, the Dwell Periods ϕ_i of user u can be a member of Registered Dataset if the user logged in to the Wi-Fi network during her visit ϕ_i.

Night Dataset: This dataset is a subset of Φ_α (*Dwell Period* of location α) where the Dwell Period happened after the closing hour and before opening hour of location α.

Definition 6 *(Night Dataset) Let t_1 and t_n be the start time and the end time of Dwell Period ϕ_i, and $\tau_{closing,\alpha}$ and $\tau_{opening,\alpha}$ be the time that location α closes and opens, Night Dataset is denoted by $\Gamma_\alpha = \{\gamma_1, \gamma_2, ..., \gamma_n\}$ where data point $\gamma_i \in \Phi_\alpha$ and $\Gamma_\alpha \subset \Phi_\alpha$ and $\tau_{closing,\alpha} \leq t_1$ and $t_n \leq \tau_{opening,\alpha}$.*

Amongst the defined datasets, the Registered (Definition 5) and Night (Definition 6) datasets (Quantitative information of these datasets for all 5 locations is provided in Table 3.2) are the key ingredients for the proposed indoor positioning system based on the two hypotheses explained at the beginning of this section. However, in order to validate the hypotheses, we collected two other data sources; *SurveyDataI* and *SurveyDataO* as are explained below.

Survey Data: To validate the initial hypotheses that whether Registered and Night Datasets are true representatives of inside and outside population, respectively, we have worked with 5 volunteers as regular customers of location A who agreed to identify their data in the Dwell Dataset Φ_A (Definition 3) to gather ground truth data. We gathered all the recent Dwell Periods (last three months) from Φ_A. The volunteers were asked to mark the records as I if the majority of their presence was inside the location A, and O, otherwise. From the surveyed data, totally, we labeled 200 Dwell Periods. They were labeled equally: 100 inside and 100 outside observations which are referred as *SurveyDataI* and *SurveyDataO* in the rest of the thesis. These 200 Dwell Periods consist of 5708 data points from Wi-Fi Dataset W_A (Definition 1) and have a variety of dwell durations.

Table 3.1: Location type and dataset description of each location

Location	Registered	Night Dataset	Reg / Night Ration	Type
A	29586	32387	0.91	1 floor
B	9398	34713	0.27	1 floor
C	5406	8708	0.62	1 floor
D	4251	53638	0.08	2 floors
E	6537	24241	0.27	2 floors

3.2.1 Noise Filter

Before the feature engineering task, in order to stabilize the RSSI values in Dwell Period sets (ϕ_i), we applied Wiener filter [68] on RSSI values. This filter is a popular noise filtering method used in a wide range of studies related to signal processing. The Wiener filter is applied to the values of each Dwell Period separately. The effect of applying this filter on the prediction performance is out of the scope of this research.

3.2.2 Feature Engineering:

In this section, we explain the features engineered from the defined datasets. The key value to identify the location of a device is the RSSI value of the probe request. Because of the multipath effect (Section 2.1) on the flying signal, the RSSI value of a *stationary* device changes frequently during a period. This is the motivation behind fingerprinting approach to classify the position of a device. In other words, the ultimate goal is to classify each Dwell Period $\phi_i \in \Phi_\alpha$ as inside or outside a store based on the characteristics of RSSI values in ϕ_i.

To achieve this goal, we engineer three time domain features from all data points in Registered, Night, SurveyDataI and SurveyDataO datasets. The resulting feature vector $\mathrm{P}_{\alpha,D}$ for each of the mentioned datasets is provided below:

$$\mathrm{P}_{\alpha,D} = \begin{bmatrix} \mu_1 & \delta_1 & \sigma_1^2 \\ \mu_2 & \delta_2 & \sigma_2^2 \\ \mu_3 & \delta_3 & \sigma_3^2 \\ \vdots & \vdots & \vdots \\ \mu_n & \delta_n & \sigma_n^2 \end{bmatrix} \tag{3.1}$$

where n is the total number of the Dwell Periods in the target dataset of location α, and D is the target dataset indicator such as Night, and Registered datasets. Below each feature is explained:

$$\begin{aligned} \mu_i &= \Big| E[\phi_i] \Big| \\ &= \left| \frac{1}{k} \sum_{j=1}^{k} O_j \right| \end{aligned} \tag{3.2}$$

Here μ_i is the arithmetic mean of RSSI values of Dwell Period $\phi_i \in \Phi_\alpha$. For each Dwell Period, δ_i is denoted by:

$$\delta_i = \Big| argmax(\phi_i) - argmix(\phi_i) \Big| \tag{3.3}$$

where δ_i is the difference of the maximum and the minimum RSSI values in ϕ_i.

Finally, the σ_i^2 is the variance of the RSSI values in vector ϕ_i which is defined as:

$$\begin{aligned} \sigma_i^2 &= Var[\phi_i] \\ &= \frac{1}{k-1} \sum_{j=1}^{k} (O_i - \mu_i)^2 \end{aligned} \tag{3.4}$$

As we have the Wi-Fi and Authentication Datasets for all 5 locations, we can generate Registered and Night datasets for all locations. However we managed to gather the ground truth datasets *SurveyDataI* and *SurveyDataO* from location *A*. Therefore, in total, we produced 12 different datasets for all 5 locations as 5 Registered, 5 Night, and 2 ground truth (SurveyDataI and SurveyDataO) datasets. Hence, the feature engineering process results in 12 feature vectors P for each dataset.

3.3 Hypothesis Validation Method

Our initial hypothesis H_R is that the device owners who log in to a Wi-Fi network at location α have spent the majority of their time inside that location (Registered Dataset). The other hypothesis H_N is that the data acquired from the environment by Wi-Fi APs at closing hours can represent the outside population (Night Dataset).

In the following sections, we aim to explain our methods to validate these hypotheses using visual, statistical and machine learning techniques.

3.3.1 Data Exploratory Analysis

Figure 3.2 shows how data is distributed in 2d space for all of the datasets of location *A*. This figure contains 3 scatter plots for all combinations of the features. In all of the three plots, Registered and Night Datasets are clearly separated for all the combinations of the features as colored in green and blue, respectively. Similarly, the survey data for both inside and outside classes are distinctly separable. The data points for SurveyDataI which are colored in yellow are in the same area as Night data points are. Similarly, SurveyDataI data points are settled in the space that Registered data points generally occupy.

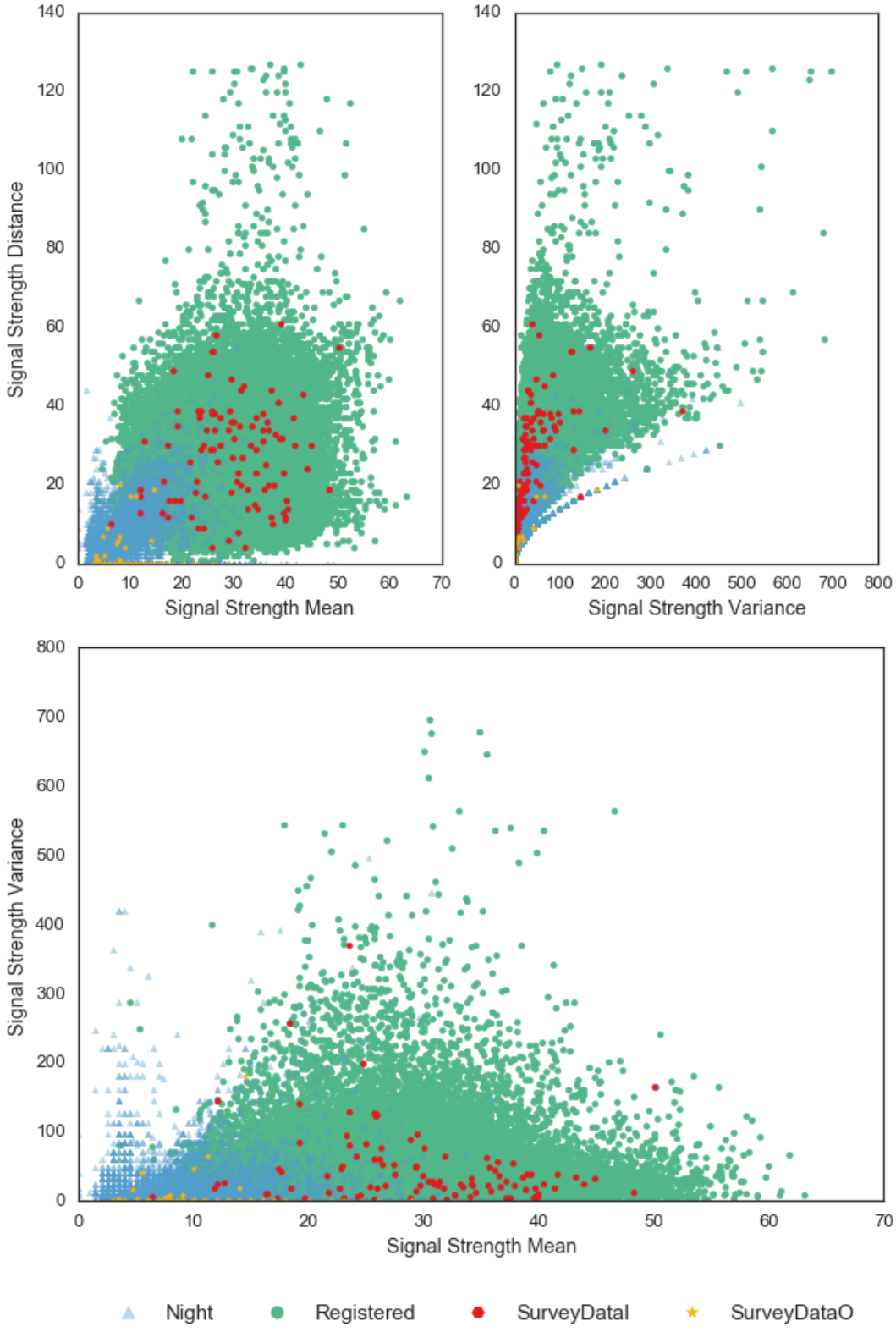

Figure 3.2: Projection of all combinations of features on a 2D space for 4 datasets of location A

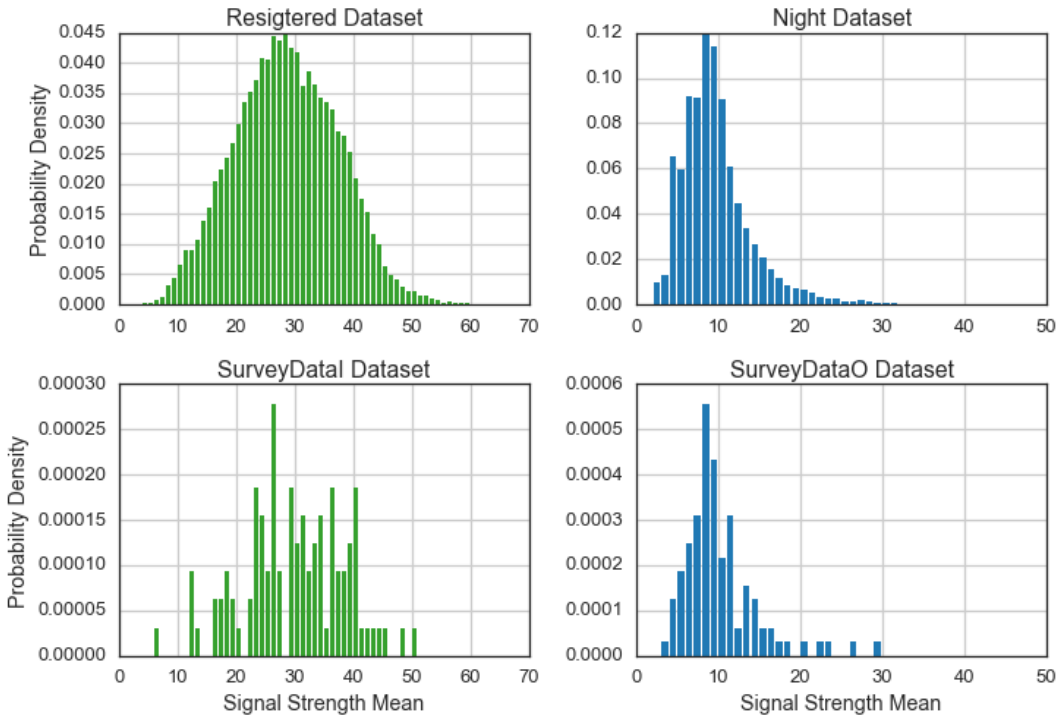

Figure 3.3: Probability density function (PDF) of RSSI Mean for all four datasets for location A

The probability density functions of location A for RSSI Mean μ of four datasets Registered, Night, SurveyDataI and SurveyDataO are shown in figure 3.3. The distribution of Registered and SurveyDataI datasets are very similar to Gaussian distribution, while the distribution of night and SurveyDataO are right-skewed. The same visual similarity exists for other features, too. As a result this initial visual analysis does not reject the H_R and H_N hypotheses.

3.3.2 Statistical Test

In this section we present the study we conducted on statistical distance between the distributions of SurveyDataI and SurveyDataO datasets and the distributions of Registered and Night datasets, respectively. For this purpose, we choose the two-sample Anderson-Darling test to quantify the distance for each feature.

The K-sample Anderson-Darling (AD) test is a non-parametric test for comparing k samples by quantifying the distance between the distribution of k samples. As a rank test, it makes no assumption on the probability distribution of the samples and

its parameters are determined based on the input samples. For the purpose of this study, we need to compare the distribution of two samples; therefore we choose the two-sample AD [53] test which is denoted by:

$$A^2_{mn} = \frac{mn}{N} \int_{-\infty}^{\infty} \frac{\{F_m(x) - G_n(x)\}^(2)}{H_N(x)\{1 - H_N(x)\}} dH_N(x) \tag{3.5}$$

where $F_m(x)$ is the empirical distribution of a random sample $X_1, \ldots, X_m$ for $m \leq x$. Here $G_n(x)$ is the empirical distribution of the second random sample $Y_1, \ldots, Y_m$ obtained from a continuous population where $n \leq x$ and $H_N(x) = \{mF_m(x) + nG_n(x)\}/N$, in that $N = m + n$ is the empirical distribution function of the combined sample. The A^2_{mn} statistic is the comparison result of the two-sample AD test to accept the null hypothesis denoted by H_0 if $F = G$ is failed to reject.

For the four datasets we perform 2 groups of test as *Registered* vs *SurveyDataI* and *Night* vs *SurveyDataO*. In each group, three individual tests are conducted for the three features. In total there are 6 individual tests accomplished in this experiment. Based on the size of both samples, the critical value for 5% significance level is 1.961 and H_0 is rejected if the A^2_{mn} statistic is higher than 1.961.

Table 3.2: 2-sample Anderson Darling Test Results

	A^2_{mn} statistic	P-value
Registered vs. SurveyDataI		
Mean	0.79630	0.154512
Variance	-0.34740	0.50148
Distance	-0.70229	0.73186
Night vs. SurveyDataO		
Mean	-0.02231	0.35630
Variance	0.47718	0.21314
Distance	0.51380	0.20536

As illustrated in table 3.2, there is no significant statistical evidence to reject H_0 at 5% level of significance where all six results are significantly lower than 1.961. It concludes that the distribution of all three features for both paired datasets are *not* statistically dissimilar. The same conclusion can be derived from the P-value as all of the results are significantly higher than 0.5 at the same level of significance. Consequently, this test fails to reject the H_R and H_N hypotheses.

3.4 Positioning System

In fingerprinting approach, a supervised method is generally used to build a model based on a training data. In fine-grained localization techniques where there are more than one spot in a location to identify (e.g. the location of each square meter should be identified), multiclass classification approach is employed to classify the observations into different classes representing different parts of a location. In this study, however, as the accuracy of interest is store level, it is adequate to differentiate observations as outside vs. inside. In this section, we explain our positioning system which is built by using Registered and Night Datasets (Figure 3.4).

As H_R and H_N hypotheses are accepted in the previous sections, we now have two labeled datasets to represent the inside and outside populations. Based on this conclusion, we draw two different scenarios as are explain in the followin sections.

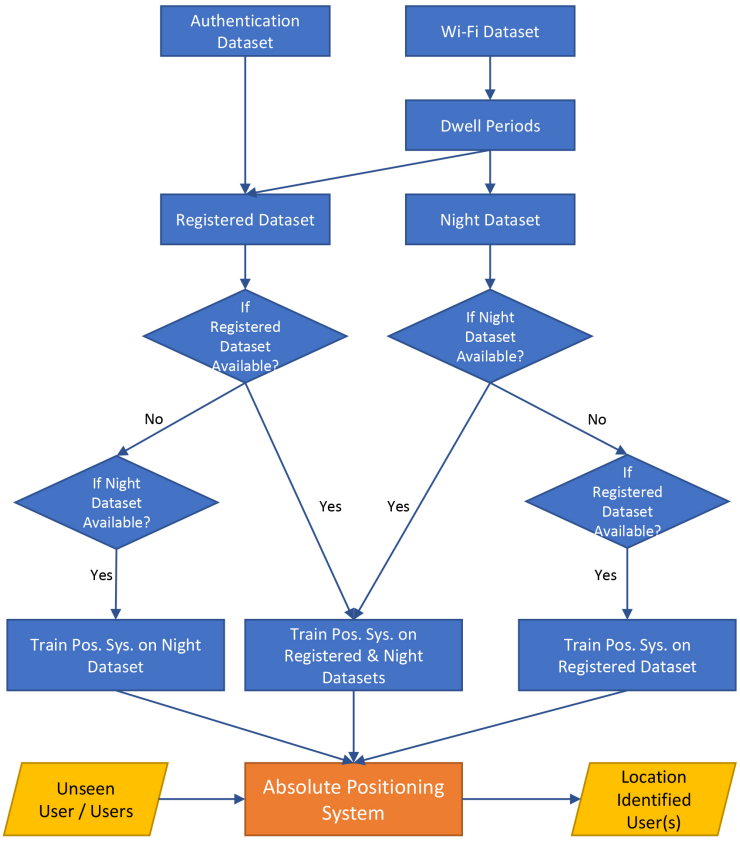

Figure 3.4: The detailed graph of the proposed positioning system

3.4.1 Two-Class Classification

Results of AD test explained in Subsection 3.3.2 show that, statistically, there is no enough evidence to fail H_R, therefore, it can be inferred that the registered dataset is pooled from a distribution very similar to the distribution of the SurveyDataI dataset. Similarly, H_N successfully passes the test. Consequently, it can be statistically inferred that night dataset is the representative of the outside population.

Considering the RSSI fingerprints of Registered and Night datasets as two *classes* for inside and outside populations, we have adequate material to build a two-class classifier to identify the location of Dwell Periods. This classifier is capable of identifying the absolute location of an unseen new device.

We employed Random Forests [37, 15] on Registered and Night Datasets to build the two-class classifier.

3.4.2 One-Class Classification:

There are some plausible scenarios in which the Night Dataset is not available. For example a restaurant where customers are served for 24 hours, 7 days a week. In addition, if night data is available, based on our knowledge about the understudied stores' neighborhood, the night data may not be a complete representative of the complete behaviour of the outside population. There are many activities that rarely happen or are totally missed at night. For instance, public transportation stops operating which affects the nature of data at certain locations close to bus stops. Offices and universities are also closed at night time. This excludes the students' or workers' activities from the night data which results in missing some unique forms of RSSI signals observed at night time. There are also some scenarios where Registered Dataset is not available if the public Wi-Fi does not require log in to access the network. In this case the Registered Dataset is not available to build a two-class classifier as explained above. Therefore, we choose the one-class focus approach to investigate both classes separately. In other words, we see how a one-class classifier performs when trained on Registered Dataset only and how it performs when modeled on Night Dataset. As a result, it shows the power of this model in handling similar situations. We employed One-Class SVM algorithm as the classifier to build the positioning system.

3.5 Discussion

As we explained in several occasions in this thesis, the proposed positioning system is an enabler of Wi-Fi Analytics as it separates the noise (outside population) from the signal (inside customers). We employed this technique as the preprocessing step for several projects in the Institute for Big Data Analytics. However, due to the scope of this thesis, we just mention the title of three projects that leveraged our positioning system as their essential part.

1. Quantifying and predicting the dwell time (duration of stay) of the customers.
2. Predict the number of customer per hour in the next 10 days.
3. Quantifying and predicting the opportunities outside of a store.

Although it is not considered in the scope of this thesis, the proposed positioning system can be used as the preprocessing stage for Partial Spatial Mining task (Chapter 5) to separate the understudied population into two groups of inside and outside population for in-depth investigation.

Chapter 4

Evaluation of Positioning System

In this section we investigate the performance of the classifier for both one-class and two-class classification approaches as the essence of the proposed positioning system. First, we investigate the performance of models built on the Registered and Night Datasets in several scenarios (Section 4.1). Then we build the proposed positioning system and with several experiments we measure its strength (Section 4.2).

For all of the experiments in the following sections, we employed Cross Validation [40] to tune the parameters of the classifiers. However, this method is used for model validation only in Section 4.2 where we employ Cross Validation to measure the performance of the models in prediction tasks.

4.1 Performance of the Classification Task

In this section, we inspect the performance of the positioning system proposed in Chapter 3 and Section 3.4 in several scenarios to show its strength in handling different situations. To achieve this goal, we conducted two group of experiments as provided below::

1. Including full dataset size in training phase and use ground truth datasets to test the classifiers (Section 4.1.1):
    - Employed the entire Night and Registered datasets to train one-class and two class classifiers.
    - Investigate the performance of the classifiers in predicting the the ground truth (SurveyDataI and SurveyDataO) datasets.
2. Including variable dataset sizes (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%) in training phase and use the ground truth datasets to test the classifiers (Section 4.1.2):

- Split the Registration and Night datasets into different dataset sizes and train the models on each data size.
- Inspect the performance of each model in predicting the ground truth (SurveyDataI and SurveyDataO) datasets.

The results of the performance tests have two outcomes. First, we can quantitatively measure the performance of the positioning system in different situations in predicting ground truth datasets. Second, results can be another indicator to accept or reject the initial hypotheses H_R and H_N.

4.1.1 Train Classifier on Entire Dataset

We first investigate the predictive capability of the model trained on Registered and Night datasets and tested on SurveyDataI and SurveyDataO using one-class classification and two-class classification techniques. Because the SurveyDataI and SurveyDataO datasets were collected from location A, for this test we only consider the Π_A and Γ_A for location A. In this experiment, we build two one-class SVM and one Random Forests classifiers. The first classifier, referred as *One-Class SVM 1*, uses Registered Dataset (Π_A) as the positive class to train the model. Then we measure the performance of the model in classifying SurveyDataI and SurveyDataO instances as inliers and outliers, respectively. The second one-class classifier named *One-Class SVM 2* is trained on Night Dataset (Γ_A) as positive class, then tested on the SurveyDataI and SurveyDataO observations. Finally, a binary Random Forests classifier is trained on both Registered and Night Datasets as positive and negative classes. Similar to other tasks, we measure the performance of this classifier based on its classification strength on SurveyDataI and SurveyDataO datasets.

The performance of the classifiers are illustrated in Figure 4.1. The performance metrics show that the classifiers are able to achieve considerably high scores for all of the experiments. A closer look at the plots, it is clear that in scenarios where both Registered and Night datasets are available, a two-class classifier can outperform the other situations by achieving 98% score for all the metrics. However, the same plot confirms that the classifier trained on Registered Dataset only can still achieve a competitive score of 95% for all of the metrics. The lowest score is for the classifier

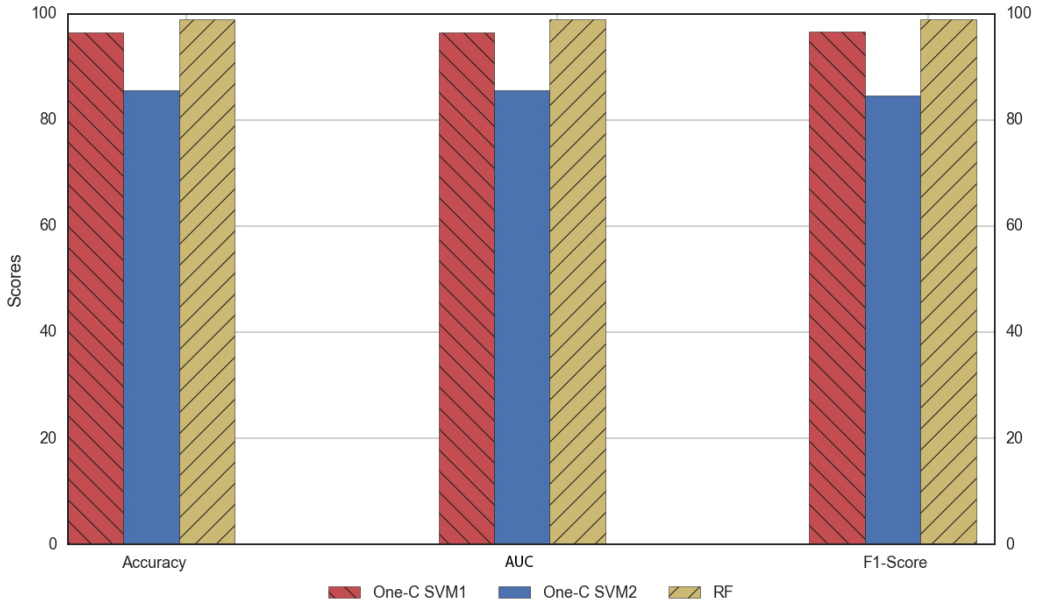

Figure 4.1: Performance of One-Class SVM1, One-Class SVM2 and Random Forests classifiers in predicting the ground truth datasets (SurveyDataI and SurveyDataO). The left bars show the comparison of the classifiers performance with Accuracy score. The middle bars compare their performance with AUC score and the right bars compare the models based on their F1-Score.

trained solely on the Night Dataset alone, however, the results show that this model can still provide reasonable score of approximately 85% for all of the metrics.

This test again validates the H_R and H_N hypotheses because a model trained by Register and/or Night datasets is able to classify the SurveyDataI and SurveyDataO datasets with considerably high scores.

4.1.2 Training on Variable Data Sizes

In this experiment, we study the performance of all three types of classifiers that are trained on different training dataset sizes and tested on the survey data. Using shuffling techniques, we shuffle and divide the data into different sizes and run the whole process 10 times. The experiment results are provided in Figure 4.2. The results show that the performance of two-class classifiers trained on both Registered or Night Datasets achieve considerably more stable and higher scores with any training sizes. These metrics are dropped significantly for both types of one-class classifications in respect to score and stability particularly with smaller data sizes (10% - 30%).

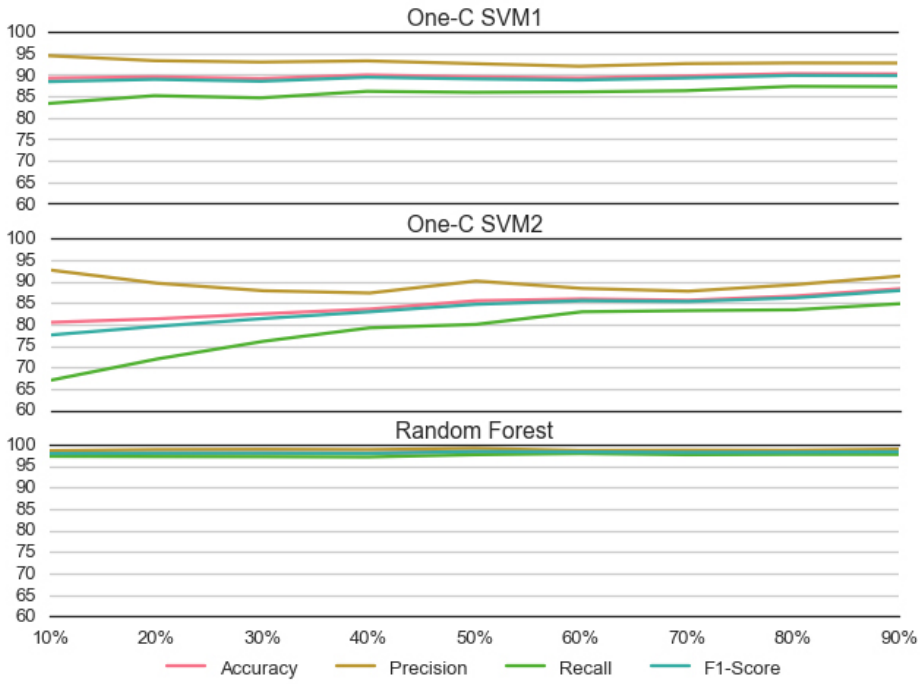

Figure 4.2: The performance results of the three types of classifiers on different training data sizes. The performance is measured based on the quality of classification on ground truth (SurveyDataI and SurveyDataO) datasets. The performance of the models is provided in Accuracy, Precision, Recall and F1-Score metrics.

However, they still show reasonably predictive capability.

The results demonstrate impressive stability and performance in different situations with small to large training sizes. In addition, these results show the acceptance of H_R and H_N hypotheses as the classifiers managed to obtain good predictive scores with different data sizes.

4.1.3 Discussion

Both one-class and two-class classifiers trained on Night and/or Registered Datasets in two conducted experiments managed to obtain satisfactory results in classifying ground truth data. Considering the 2-sample AD test results (Section 3.3.2), we can conclude that Registered and Night Datasets are very similar to SurveyDataI and SurveyDataO datasets. Hence, they can be representative of inside and outside populations. However, classifiers trained solely on the Night Dataset showed that Night Dataset is not as powerful as the classifier trained solely on Registered Dataset

or on both datasets.

4.2 Performance of the Positioning System

In this section, we inspect the performance of the positioning system in several scenarios. We build a positioning system with both assumptions using one-class and two-class classification methods. Then, we evaluate and compare the ability of all approaches in classifying positions of Dwell Periods from different angles (Section 4.2.1). Next, we investigate a transfer learning approach, where a model is trained on one location, and evaluated based on the predictive capability of the model in classifying Dwell Periods of other locations (Section 4.3).

4.2.1 Model Validation with Cross Validation Test

After achieving reasonable results in the previous section (Section 4.1), we concluded that the positioning system has reasonable performance in all three scenarios in classifying ground truth datasets in addition to other two tests (Section 3.3) we concluded that *Registered* and *Night* Datasets are reliable representatives of inside and outside populations, respectively. Hence, these datasets can be used as the labeled datasets to build the absolute fingerprinting positioning system. In this section, we aim to validate the performance of the positioning system built by the two datasets. We conducted three experiments to inspect the performance of the positioning system in all three scenarios: 1) availability of Registered Dataset only, 2) availability of Night Dataset only, 3) availability of both Registered and Night Datasets. Then, using k-fold Cross Validation technique, we validate the models.

We performed three experiments on the classifier trained on Registered and Night Datasets using k-fold Cross Validation. For the one-class classification we train the model on one of the datasets (Registered or Night Datasets) using 10-fold Cross Validation where the model is trained on 9 folds and then validated by the remaining 1 fold. However, because of the essential limitation of one-class classification, in validation set we include the other dataset in the validation set. For example, considering One-Class SVM 1 that is trained on Registered Dataset, we build the model on 90% (9 folds) of the Registered Dataset and validate the model on the remain 10% (remaining 1 fold) of the Registered Dataset in addition to the entire Night Dataset.

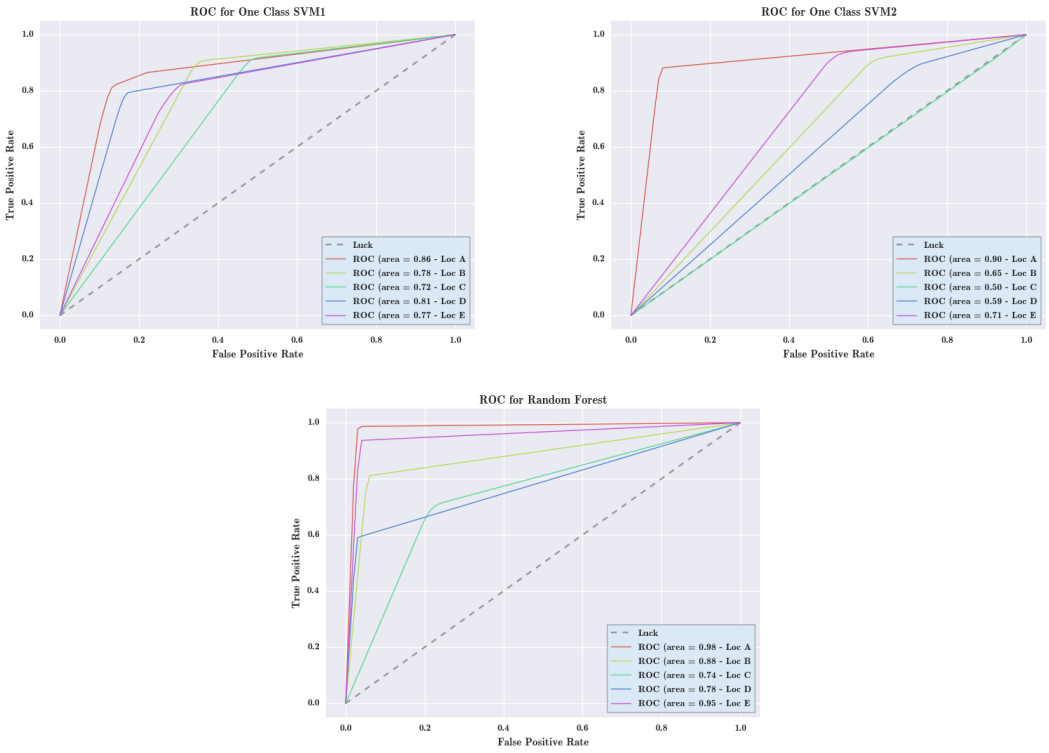

Figure 4.3: The ROC AUC graphs of the positioning system in all three scenarios: 1) One-Class focus - availability of Registered Dataset only, 2) One-Class focus - availability of Night Dataset only, 3) Two class focus - availability of both Registered and Night Datasets

This process is repeated for both datasets, separately. This experiment is slightly different for two-class classification task where in this approach we build the model on 90% (9 folds) of the both Night and Registered Datasets and validate the model on the remaining 10% (1 fold) of the entire datasets. The entire process for both one-class and two-class approaches are repeated 10 times for all 5 locations. The results of this experiments are provided in Figure 4.3 using ROC AUC graph.

The results show that Random Forests outperformed the other classifiers with reasonable AUC scores for all five locations. However, except one location (Location C) in One-Class SVM 2, the results of other classifiers show reasonable performance. Similar to other experiments, we can conclude that the performance of the positioning system increases significantly when both datasets are available.

Table 4.1: AUC Score of mapping a model from one location to the rest of the locations

Trained on	Algorithms	Tested On					Mean
		A	B	C	D	E	
Location A	OC SVM1		**0.83**	0.61	0.83	0.95	0.81
	OC SVM2		0.79	0.57	0.73	0.87	0.74
	RF		0.81	**0.71**	**0.90**	**0.96**	**0.84**
Location B	OC SVM1	**0.85**		**0.74**	0.66	**0.77**	**0.76**
	OC SVM2	0.64		0.56	0.54	0.61	0.59
	RF	0.78		0.67	**0.75**	0.76	0.74
Location C	OC SVM1	**0.97**	**0.69**		**0.84**	0.94	**0.86**
	OC SVM2	0.84	0.51		0.61	0.85	0.70
	RF	0.95	0.60		0.83	**0.95**	0.83
Location D	OC SVM1	**0.96**	**0.88**	**0.68**		**0.94**	**0.87**
	OC SVM2	0.74	0.63	0.51		0.68	0.64
	RF	0.84	0.67	0.61		0.84	0.74
Location E	OC SVM1	**0.97**	**0.83**	0.61	**0.84**		**0.81**
	OC SVM2	0.73	0.68	0.53	0.63		0.64
	RF	0.93	0.75	**0.62**	0.83		0.78
Mean		**0.85**	0.72	0.62	0.75	0.84	0.76

4.3 Evaluate the Positioning System in Transfer Learning

Transfer Learning is to improve the performance of a learning task by transferring the knowledge from another by related task that is already learnt [55, 16, 72]. Motivated by this method, in this section, we investigate performance of the proposed indoor positioning system in transfer learning situation. In other words, we inspect how accurate a model trained on datasets of location α can classify Dwell Periods of location β.

Here, we present the strength of model transferability for all locations where we evaluate the positioning system built on datasets of location α by measuring its performance in classifying the Dwell Periods of other 4 locations. The main purpose of this experiment is to study the feasibility of mapping a model from one location to another. This result is appealing in particular scenarios where neither of Night nor Registered Datasets are available or the labeled data is very limited.

As table 4.1 suggests, with the availability of only Registered Dataset, in average, we can confidently achieve a high score (AUC 82%) in predicting the data of other locations using one-class classification. However, unlike other experiments, availability of both datasets results in slightly lower scores in almost all experiments compared to other scenarios as Random Forests classifier obtained the second best results in transfer learning with 0.79 AUC score in average.

4.4 Discussion

In this chapter we evaluated the positioning system using several experiments and provided the results. In general, we found that the Registered and Night datasets can be true representatives of inside and outside populations, respectively. In addition, we discovered that in the lack of one of those datasets, we can still achieve a reasonable performance for the positioning system. However, in most of the results we found that the ideal case is to acquire both datasets to build the positioning system. Finally, we showed that the positioning system is reliably transferable to other locations in case the labeled datasets are unavailable. This positioning system can be used at the preprocessing stage of a Wi-Fi Analytics platform to identify the absolute position of the users.

The performance of the classifiers trained on both the Registered and Night Datasets had much higher performance than the performance of the classifiers trained on only one of the datasets. The main reason is the additional information that is available to the classifier in the training phase. On the other hand, the one class classifiers have only one side of the information: the inside or the outside population. Therefore, the positioning systems built solely on one of the datasets have lower predictive capability than the one trained on both datasets.

Chapter 5

Mining Partial Spatial History

In this chapter, we present our method to extract spatial behavioral patterns from a group of people from their SSID data. The first section explains the problem statement and the challenges, then we explain the necessary preprocessing steps and finally we explain our proposed method.

5.1 Problem Statement

A probe request propagated from a device, contains several fields such as device MAC address, RSSI value, time of packet propagation and the SSID. The SSID field contains a Wi-Fi network name (SSID) that refers to a network the device was previously connected to. For instance, if you have been connected to the Wi-Fi network of café α with SSID $Network\_\alpha$, then your phone saves the name in a list called Preferred Network List (PNL). PNL in each smart device contains the names of the previously connected networks. In the next visit to location α, the smart device includes one of the SSID records from the PNL into the probe request. This is a normal mechanism of connection reestablishment to reduce the network search process in roaming situation. SSIDs leak some information as they are simply human readable [51]. Mapping this coarse-grained spatial information to its actual location can unveil valuable insights into the smart device holder. This spatial data is more machine readable. Therefore, collecting the SSIDs from the probe request of a large population can provide deep insights into a population because each SSID can be treated as spatial information about the owner of the device. The type and the geographical information of the location tells us some information about the interest, nationality, ethnicity, and social class of an individual. For example, if the collected SSIDs of a large population who attended a conference include SSIDs from country β, we can infer that the attendees and the conference have some links to this country. In addition, if the SSIDs of the population contain a large number of SSIDs related

to a university, it can be inferred that a percentage of the population attended the university and probably have university degree. These are some examples to show how the history of visited locations can provide interesting information about an individual or a population. Collecting a large number of SSIDs from a population, provides a number of inferences about the population. Considering the inferences as the features which are extracted from a given a set of SSIDs T for population U, we are interested to group the population into k groups based on the features[1]. In this chapter, we aim to present a framework to find similar groups of people from the collected SSIDs. An overview of our proposed framework is plotted in Figure 5.1.

This task, in particular, is priceless for brick-and-mortar businesses as they can understand the types of their customers in a non-intrusive and inexpensive fashion. Considering a small or medium size brick-and-mortar business by leveraging this technique is able to extract the history of previously visited locations of their customers at their first visit without paying the price of explicit tracking systems. In other words, the alternative way to obtain such information is to track the individuals by having an app installed on their smart devices or purchasing such data from mobile network operators which can cost from thousands to millions of dollars [6].

To reach the aim of this chapter, there are two important but challenging phases that should be passed before reaching the clustering step. First, the SSIDs should be mapped to an actual location. Second, the spatial behavioral features of population based on the mapped locations should be extracted. Then, individuals can be clustered based on the extracted features. In the rest of this chapter we address each in detail.

5.2 Map SSIDs to Actual Locations

In this section, we preset our method to map SSIDs to their actual locations. For example, we want to map SSID name *CoburgCoffeeWiFi* to *Coburg Coffee House* located at '6085 Coburg Rd, Halifax, NS B3H 1Z3'. But first, let's explain characteristics of SSID data that makes this mapping process challenging:

- **Duplication of SSID names**: Unlike MAC address, SSIDs are not unique.

[1]In this thesis, we are not interested to make individual level inferences, instead, aggregate of the data is appealing to us

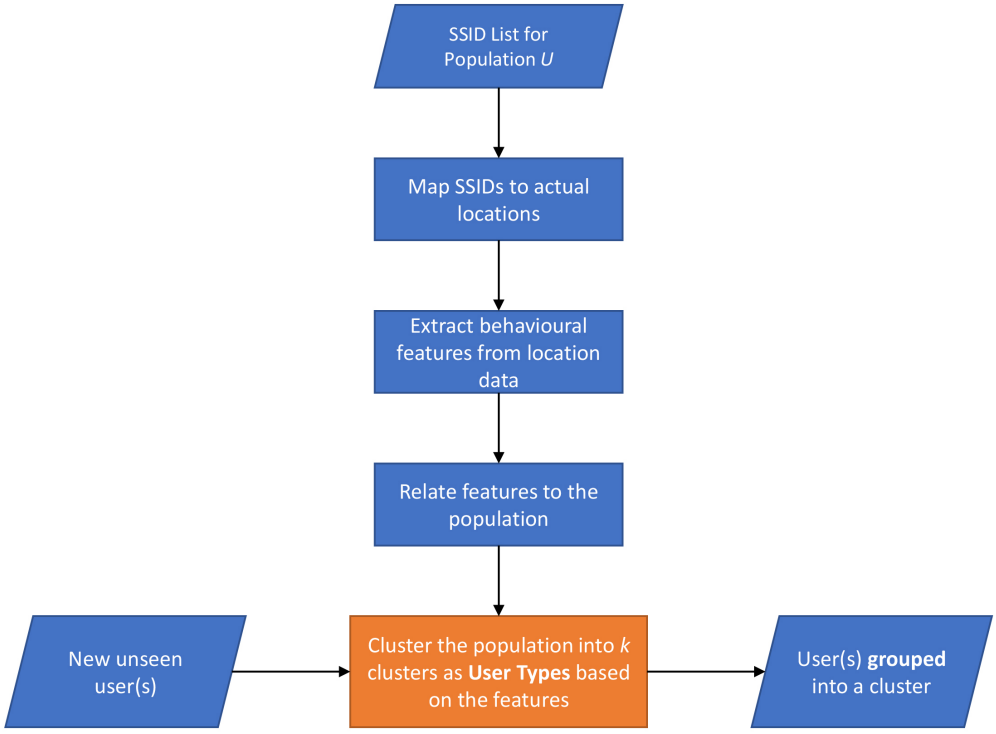

Figure 5.1: An overview of the SSID-based population clustering framework

Therefore, there may be several Wi-Fi networks with the same SSID name. For instance, anyone is allowed to name her Wi-Fi network $TimHorton's$ even if there is no relation between her and Tim Horton's Café. In addition, it is possible that all of the branches of a chained store like $Walmart$ use a unique SSID name ($walmart$) for the entire of their store chain. Therefore, in this case, it is infeasible to identify the exact location of the SSID. Instead, this SSID indicates that this person most probably visited a Walmart in the past.

- **Unclear number of visits**: The hidden number of visits is unclear from the SSID record of a probe request. Retrieving the SSID S form a device only confirms that the device owner, visited a location with SSID name S at least once. However, it is not clear how many times she visited the location in the past.

- **Lack of temporal information**: The SSID field does not contain any temporal information about the connection event. In other words, there is no field

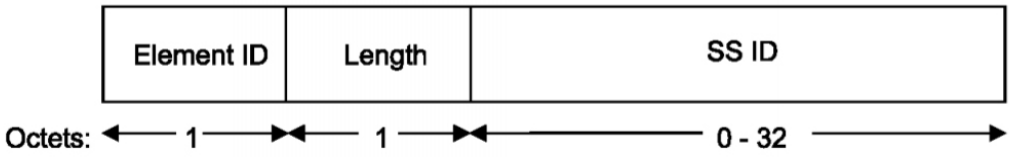

Figure 5.2: The SSID section in a probe request packet

in the probe request that shows a temporal information about the connection event. Therefore, we can only infer that the visit happened in the past with no precise temporal information.

To alleviate these challenges we first describe the nature of the collected data and the preprocessing steps taken to ease the mapping process.

5.2.1 Mapping Approaches

SSID data is extracted from probe request packets. This information is a string with maximum of 32 characters that consists of ASCII letters (Figure 5.2)[5].

We collected SSIDs in three events at two locations including Goldberg Computer Science Building[2] and the office of SolutionInc Limited[3] that both are located in Halifax downtown using Paspberry PI 3 B+ programmed with Python 2.7 and equipped with a long range USB WiFi wireless adapter. In total, we recorded the probe requests of 7053 individuals (unique MAC addresses) and 3340 unique SSIDs. The reason for obtaining less SSIDs than the total number of observed devices is that 1) many of the SSIDs are shared amongst many devices (Figure 5.2) and 2) the majority of devices unveiled zero[4] - one SSID through their probe requests (Figure 5.4).

In order to map the SSID to a location, the related research work that are explained in Section 2.2, used two different methods 1) Using Named-Entity Recognition (NER) techniques [19] and 2) Online Services [48, 20, 19]. Based on the cited research work, online services provide better results compared to the NER techniques.

[2]Located at: 44.637409, -63.587189

[3]Located at: 44.657883, -63.596461

[4]The SSID field in some of the probe requests is empty and the author could not find a reliable source to explain the reason of this behavior

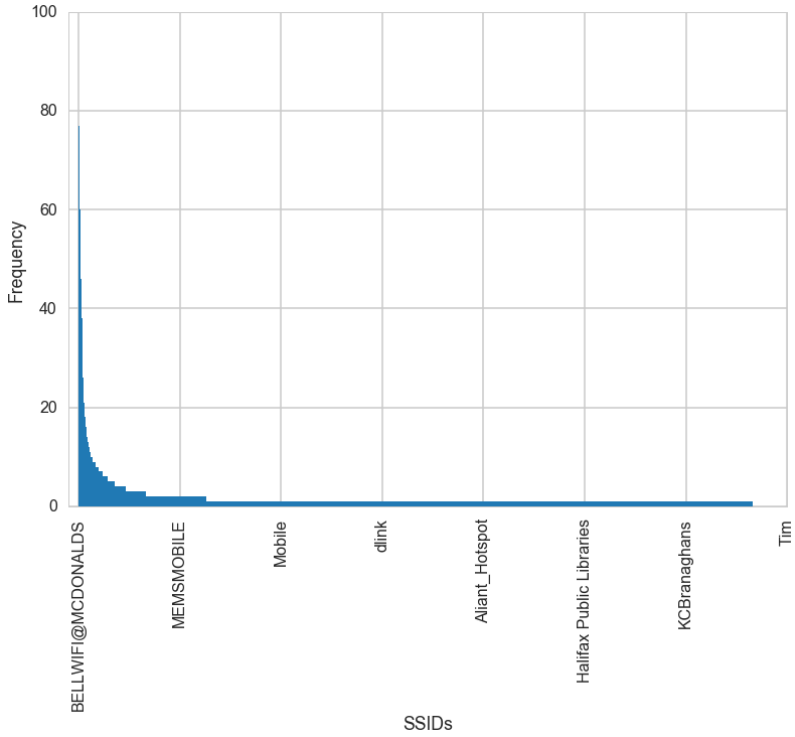

Figure 5.3: The frequency of SSIDs collected from the probe request packets. Each point in x axis corresponds to an SSID. In order to plot a comprehensible visualization, we removed the most frequent SSID (*BELL_WIFI*) with 3546 appearance from the graph. Therefore, this plot shows the SSID occurrence from the second most frequent SSID (left hand side) to the least frequent SSID (right hand side)

For a number of reasons, amongst the geo-location online services [5], Wigle.net and Google Places API gained more popularity. A comparison of these services is provided below.

Wigle: Wigle.net is a popular API to resolve the location of SSIDs in a majority of the related work[48, 20, 19]. The Wigle is a crowdsourced database filled by volunteers who have Wigle's phone app installed on their smart phones. This app reports all of the reachable Wi-Fi networks SSIDs with the current GPS coordinates of the phone. This introduces the first error as a Wi-Fi network can be several meters far from the phone while the registered coordinates in the database is the coordinates of the smart phone. Therefore, the geometric information of the SSID is not reliable

[5] 1) Skyhook → www.skyhookwireless.com
2) Wigle → https://wigle.net
3) Microsoft Bing Search API → https://datamarket.azure.com/dataset/bing/search
4) Google Places API → https://developers.google.com/places/

and cannot be easily mapped to the actual location. The other essential error of this service is its disability to remove non-existing SSIDs. In other words, many of the reported Wi-Fi network names do not exist in that location. Wigle returns the entire results for a given SSID. This is another reason of uncertainty as it does not remove the abolished locations. The other limitation is that it does not provide any semantic information about the location. And finally there is a 100 query limit per user per day which makes it very impractical.

Google Places API: Google Places API is a service to search the Google database for a specific location by a keyword like *TimHortons*. It almost addresses the majority of the limitations and errors Wigle possesses. First, it provides the exact location of an SSID given the SSID name is similar to the location name. In addition, Google Places repeatedly renews and validates its search results[33] to avoid returning out-dated information. The returned results contain several semantic information about the location such as location types, address, ratings (if applicable), logo, screen shot. Finally, Google API provides a generous package to request 150,000 queries in a day for free. Google Places API also provides some features which are very useful for retrieving better results. SSID names have several wild characters, non-split words and typos. This API internally solves all of these potential issues with automatic

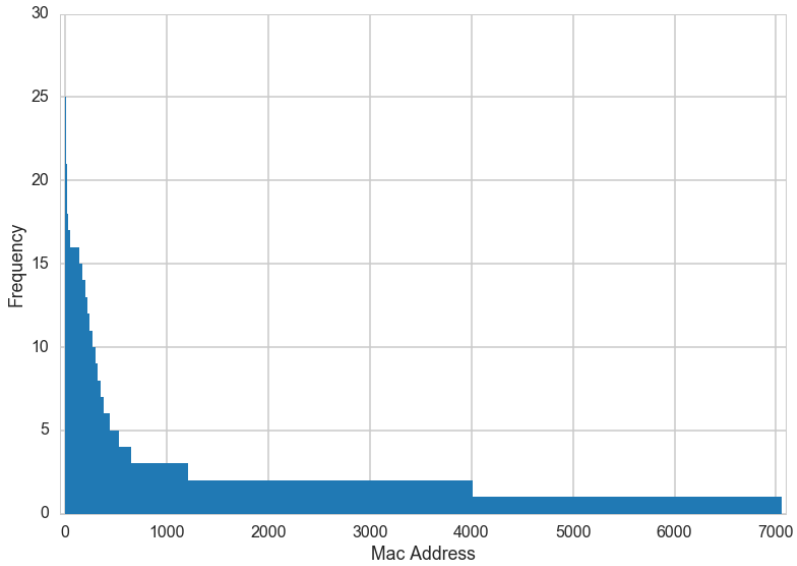

Figure 5.4: The number of SSIDs captured from the devices where each point in x axis corresponds to a device. In total there were 7053 devices.

word breaker and spell correcter to make SSIDs search friendly. It also enables us to search based the relevancy to a location or coordinate termed *location constraint*. As we collected the entire SSID data from Halifax downtown, we employed this feature to sort the results based on their distance to Halifax downtown. By setting the location constraint to Halifax Downtown, we assume that the first item of the response object is the most relevant amongst others.

Although Chernyshev et al. [19] proposed that Wigle API provides better results than Google Place API, because we are interested in semantic information about the locations to cluster the population based on their interests, and considering that Google Places API provides almost all the required information, we preferred Google Search API over the other alternatives.

5.2.2 Discovering Semantic and Geographical Information

We searched the entire 3340 SSIDs using Google Search API with considering Halifax Downtown as the *location constraint*. Each item in a search result contains several fields. Amongst them we are interested in location address ($formatted\_address$), coordinates ($geometry$), location name ($name$), and location types ($types$). The selected fields provides related data that is used in the feature engineering section (Section 5.3.1). A sample result is provided in Appendix A.1.

Search results: Most of the brick-and-mortar businesses choose a descriptive name for their SSIDs. However, we found that residential places have very similar names with no actual meaning. For example, residential places that are Bell Customers have very similar SSID names that start with *BELL-WIFI*. Google Places, mistakenly maps these SSIDs to the office of the internet provide company that in this case is Bell Canada. In addition, there are some SSID names that have no meaning from geographical point of view like a sequence of random numbers or a Twitter user account. We found that there are 255 frequent SSIDs that have such characteristics. Because we are interested in finding the interests of the people based on their spatial history, a residential location can not provide related information to help our objective. As a result, we removed these locations from the entire dataset.

This mapping process unveils initial yet interesting results as presented in Figure 5.5, 5.6 and 5.7. These plots present initial insights about the observed population.

Figure 5.5: The geographical distribution of SSIDs throughout the world. The size of the bubble shows the number of users observed at these locations. The color intensity indicates the the distance from the data collection point and the location coordinates (Definition 14).

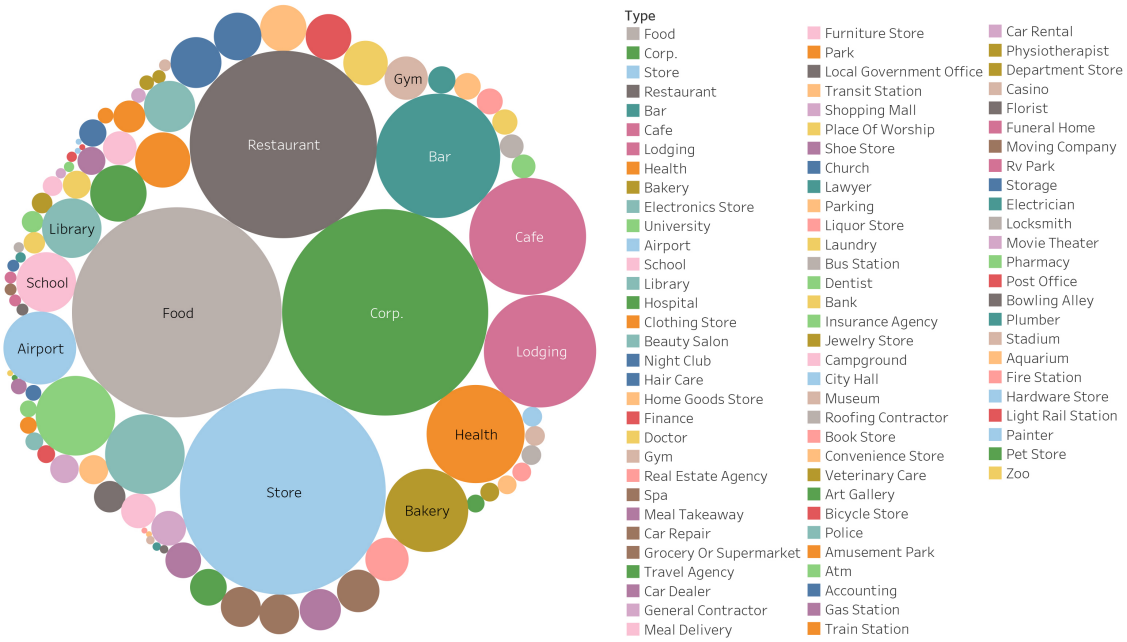

Figure 5.6: The entire location types extracted from the captured SSID. The bubble size corresponds to the frequency of the location type y (Definition 12) $w.r.t$ the number of devices observed at locations with location type y.

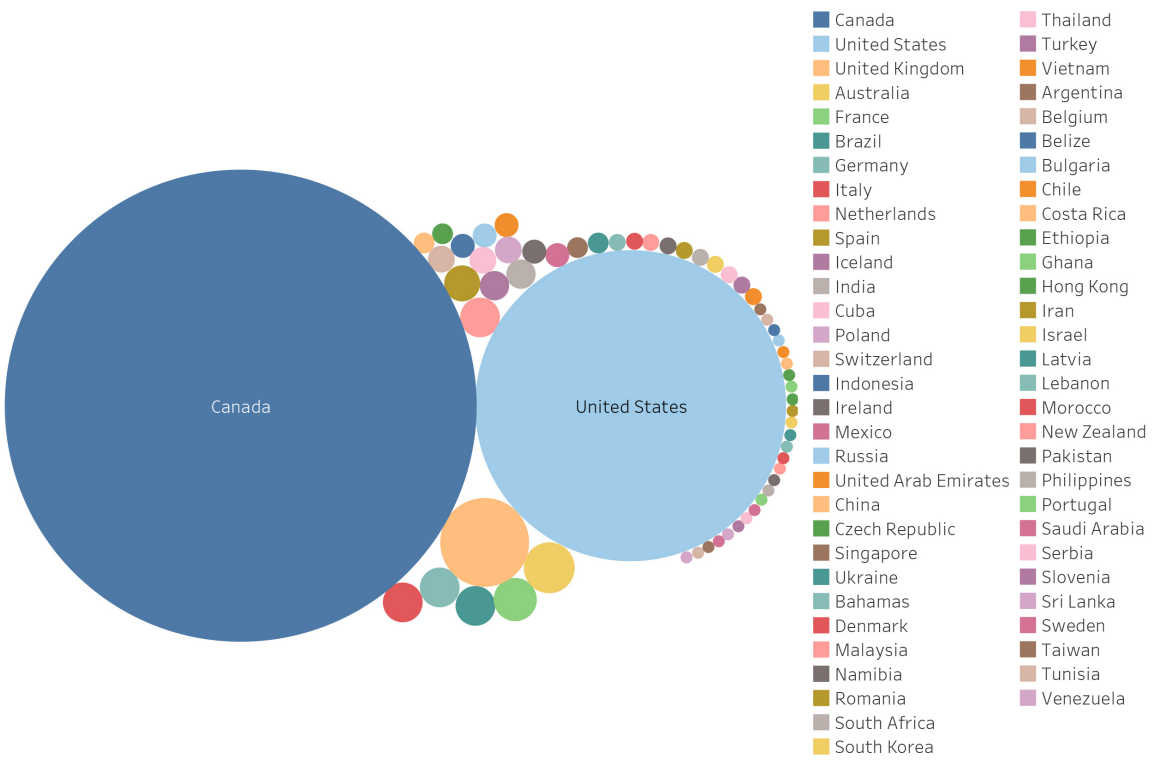

Figure 5.7: The countries of the captured SSIDs. The size of the bubbles corresponds to the frequency of a country c $w.r.t$ the number of SSIDs with the country name c.

Such as the most frequent places, countries, and location types that the understudied population visited in the past. There are some business questions that can be answered by plotting these graphs such as popular location types, ethnicity, and nationality. For example, from 5.5 and 5.7 we can infer that the majority of people have come from Canada and US. However, there are several SSIDs that highlighted Europe, and East Asia. Moreover, it shows that members of this population have visited several countries in all continents. We can also infer that people visited east and west of the US in addition to the major Canadian cities.

5.3 Clustering the Population Based on their SSID Data

In this section, we present our method to cluster individuals based on the extracted initial information explained in Section 5.2 in addition to the engineered features that is explained in Section 5.3.1. Later we describe our method to cluster SSID data to group the observed people into similar clusters based on their spatial behaviours.

5.3.1 Feature Engineering

In this subsection we explain our feature engineering technique necessary for clustering the observed population based on their trajectories. First, amongst the retrieved information explained in the previous section, we extract 4 features to describe a location as location type, the distance between the location and the data collection center, location entropy, and location user count. Then, from these features, we generate several user related spatial behavioural features to define the a user. We first start with the traditional definition of a trajectory.

Definition 7 *(Trajectory) A Trajectory is a finite set* T_u = *(*(x_1, t_1)*,* (x_2, t_2)*, ...,* (x_n, t_n)*) where each point* x_i *indicates a multidimensional feature vector of a visited location and* t_i *indicates the time of the visit of the moving object (user)* u*.*

The level of granularity of trajectories can be very diverse from any movement of a moving object in decimeter level granularity to visited points of interest (POI) in a continent. In addition, the temporal aspect of trajectory data can be represented as multidimensional time series [43] or multidimensional data points [54]. On the other hand, SSID data are special form of trajectory data as temporal information as for each data point in trajectory set visited time t is unknown. Hence, we define a modified definition for SSID-based trajectory and name it **Partial Spatial History**. In the rest of this thesis, we refer SSID-based trajectory as Partial Spatial History.

Definition 8 *(Partial Spatial History) Partial Spatial History is a finite set* T_u = *(*(x_1)*,* (x_2)*, ...,* (x_m)*) where each data point* x_i *corresponds to the multidimensional feature vector of a point of interest (POI) with no temporal information for user*

In other words, a data point x_i corresponds to one SSID name extracted from the device of user u. Unlike normal trajectories, Partial Spatial History is coarse-grained and inexact as the data is sparsely collected. For instance, we may capture three unique SSIDs from probe requests of user u that each of them may belong to a different continent while normal trajectory data can be as fine-grained as decimeter movements of a moving object. However, it **does not** necessarily mean that this user only connected to three other networks before.

Definition 9 *(SSID Database) Let U be the list of observed users, and $u \in U$ be an individual user, there is a vector $D = (T_{u_1}, T_{u_2}, ..., T_{u_n})$ where data point T_{u_i} corresponds to the Partial Spatial History vector of user u_i. Data points in vector D have the following characteristics:*

- The length of T_{u_i} is denoted by $\left|L(T_{u_i})|\exists T_{u_i} \exists T_{u_j} where L(T_{u_i}) \neq L(T_{u_j})\right|$

The number of captured SSIDs is proportional to the length of the proximity of a device to the capturing antenna and the number of saved SSIDs on the device. Hence, the length of data points in vector D varies from one user to another. This results in different lengths of SSID list for each individual.

$$\mathrm{D} = \begin{bmatrix} x^1_{u_1} & x^2_{u_1} & \dots & x^o_{u_1} \\ x^1_{u_2} & x^2_{u_2} & \dots & x^n_{u_2} \\ \vdots & \vdots & \ddots & \vdots \\ x^1_{u_k} & x^2_{u_k} & \dots & x^m_{u_k} \end{bmatrix} \tag{5.1}$$

Database D contains k rows as the number of individual users $u \in U$ and a variable length column size that is different from one user to another depending on the number of captured locations (SSIDs). Each data point $x^i_{u_j}$ corresponds to a multidimensional feature vector of location x_i in Partial Spatial History vector T_{u_i}. One group of features corresponds to the type of the location that is retrieved from Google Places API.

Definition 10 *(SSID Types Vector) Let $Y_x = (y_1, y_2, ..., y_n)$ be the SSID Types Vector for location x where an item y_i is a type that explains one aspect of location x. Data point y_i has the following characteristic:*

- The length of y_i is denoted by $\left|L(y_i)|\exists y_i \exists y_j where L(y_i) \neq L(y_j)\right|$

The length of retrieved types for each location can be different from one SSID to another. Therefore, based on the $L(y_i)$ the length of feature vector varies for each observed location.

In addition to the location types, we produced three other features for each location x named *Entropy* denoted by $Entropy(x)$, *Distance* denoted by $Distance(x, O)$, and *UserCount* denoted by $UserCount(x)$.

Let y be a type, U be the list of all users and u is a user where $u \in U$, O_u is the set of types where u was visited and $O = \cup_{u \in U}$ and $o \in O$ is a two-tuple entity $o.MAC, o.Type$. Let $U_y = \{u \in U \mid u \text{ was observed at type } y\}$, $O_{u,y} = \{o \in O : o.Type = y\}$ is the set containing observation of user u at locations with type y and $O_y = \{o \in O : o.Type = y\}$ is the set of observations that happened for type y regardless of the user u. The probability that in a random draw u belongs to O_y is $P_y(u) = \frac{|O_{u,y}|}{|O_y|}$. In words, $P_y(u)$ is the total fraction of all records of user u at locations with type y.

Definition 11 *(Type Entropy) Type entropy is denoted by* $Entropy(y) = -\sum_{u \in U_y} P_y(u) \log P_y(u)$ *which quantifies the homogeneity of the type y w.r.t the variation of observed users u.*

In words, *Type Entropy* shows how much a location type is homogeneous $w.r.t$ the users. For example, if the locations with location type y were visited by completely different users, $Entropy(y)$ obtains a high value.

Definition 12 *(Type Frequency) Let Freq(y) = $|O_y|$ be the frequency of type y which indicates the total number of times a location with type y was visited.*

In words, *Type Frequency* shows how frequent locations with location type y were visited regardless of the homogeneity of users. For example, if locations with type y were visited several times solely by user u, $Freq(y)$ obtains high value.

Considering U_y to be the list of types that each individual user visited, the other feature is the total number of unique users defined as:

Definition 13 *(User Count) Let $UserCount(y) = |U_y|$ be the number of unique users (devices) observed in locations with type y.*

Definition 14 *(Location Distance) Given the coordinates of location x and data collection point C, Location distance $Distance(x, C)$ is the geographical distance between $x.Coordinates$ and $C.Coordinates$.*

Location Distance is the geographical distance between the coordinates of location x and the coordinates of data collection centers (Goldberg Computer Science Building

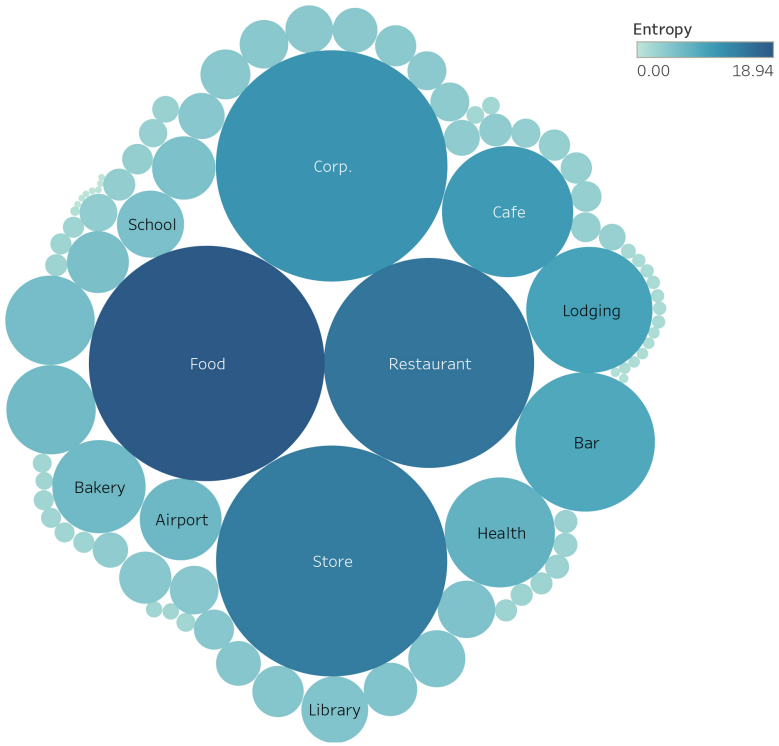

Figure 5.8: The entropy of location types *w.r.t* the observed users. The size of the bubble is the frequency of the location types while the color intensity shows the entropy.

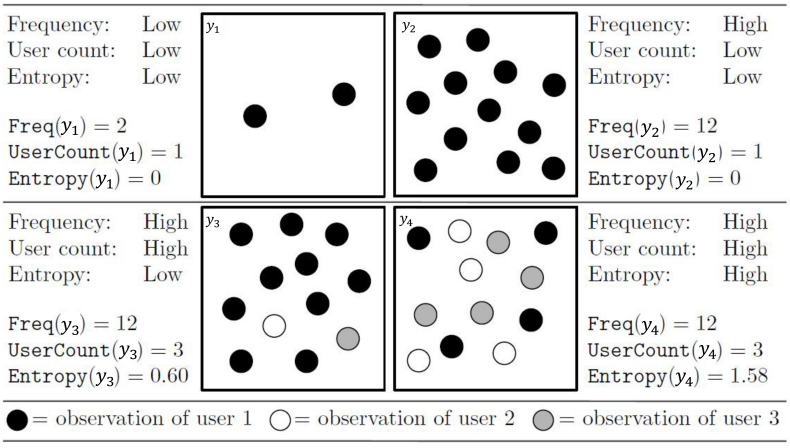

Figure 5.9: The entropy, frequency and the user count features for a four hypothetical types (y_1, y_2, y_3, and y_4) and three users. This plot is inspired from the work of Cranshaw et al. [23].

and SolutionInc Office). As we collected data from two sites, we consider the related data collection center for each location to calculate this feature.

In order to find the distance between locations, we consider geographical distance instead of other distance functions because of the spherical shape of earth. The geographical distance equation is provided below:

$$hav(\theta) = \frac{1 - cos(\theta)}{2} \tag{5.2}$$

$$h(\varphi_1, \varphi_2, \lambda_1, \lambda_2) = hav(\varphi_1 - \varphi_2) + cos(\varphi_1)cos(\varphi_2)hav(\lambda_1 - \lambda_2) \tag{5.3}$$

$$d(\varphi_1, \varphi_2, \lambda_1, \lambda_2) = r2\arcsin(\sqrt{h(\varphi_1, \varphi_2, \lambda_1, \lambda_2)}) \tag{5.4}$$

where φ_1 and φ_2 are latitude of point 1 and 2, λ_1 and λ_2 are longitude of point 1 and 2, $h(P_1, P_2)$ is the *Haversine formula* [59] for points P_1 and P_2, r is the radius of the sphere which is 6371e3 meters for earth, and finally, $d(P_1, P_2)$ is the geographical distance between P_1 and P_2.

From the retrieved address provided by Google Places API, the country is also extracted as the last location related feature.

Definition 15 *(Country) The country of location x is defined as $Country(x) = x.Country$.*

The total extracted features for location x are brought in Equation 5.5 and Figure 5.10:

$$\begin{aligned} x =< Y_x, &Entropy(Y_x), UserCount(Y_x), Frequency(Y_x), \\ &Distance(x, C), Country(x) > \end{aligned} \tag{5.5}$$

where Y_x is the set of types for location x, $Entropy(Y_x)$, $Frequency(Y_x)$ and $UserCount(Y_x)$ are the set of entropy, frequency and user count, respectively, for the location x with types Y_x. $Distance(x, C)$ is the distance of x to C and $Country(x)$ is the country of location x.

Considering Definition 8, and Equations 5.5 and 5.1, for each user u there is a variable length feature vector T_u that for each item x_i in T_u there is a multidimensional feature vector that defines location x_i. Therefore, for user u, the features that defines the spatial behaviours of user u are the concatenation of the feature vectors of locations in T_u. in other words, for user u the feature vector is defined as follow:

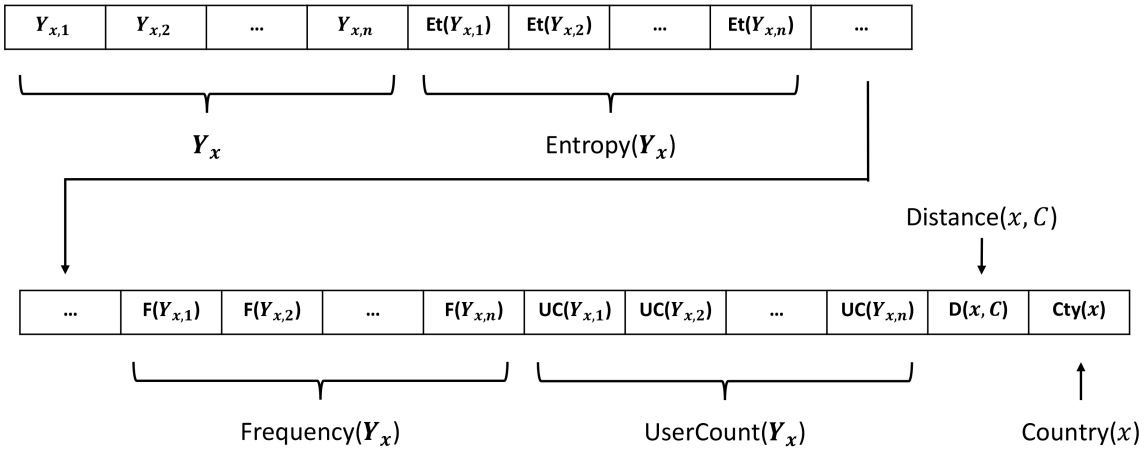

Figure 5.10: A graphical representation of extracted features for location x.

Definition 16 *(User Types Vector) For user u there is a vector $Y_u = (Y_{x_1}, Y_{x_2}, ..., Y_{x_n})$ as user types vector that each item Y_{x_i} consists of types of location x_i.*

Definition 17 *(User Types Entropy) For user u, there are four features calculated as $argmin\, Entropy(Y_u)$, $argmax\, Entropy(Y_u)$, $\mu(Entropy(Y_u))$, and $\sigma^2(Entropy(Y_u))$ are the **min**, **max**, **mean** and **variance** of Type Entropy vectors of Y_u for user u.*

Definition 18 *(User Types User Count) For user u, there are four features calculated as $argmin\, UserCount(Y_u)$, $argmax\, UserCount(Y_u)$, $\mu(UserCount(Y_u))$, and $\sigma^2(UserCount(Y_u))$ are the **min**, **max**, **mean** and **variance** of Type UserCount vectors of Y_u for user u.*

Definition 19 *(User Types Frequency) For user u there are four features calculated as $argmin\, Frequency(Y_u)$, $argmax\, Frequency(Y_u)$, $\mu(Frequency(Y_u))$, and σ^2 $(Frequency(Y_u))$ are the **min**, **max**, **mean** and **variance** of Type Frequency vectors of Y_u for user u.*

Definition 20 *(User Visited Locations Count) For user u $LocationCount(u) = |T_u|$ is the number of locations user u visited.*

Definition 21 *(User Visited Locations Distance) For user u there are four features calculated as $argmin\, Distance(T_u, C)$, $argmax\, Distance(T_u, C)$, $\mu(Distance(T_u, C))$, and $\sigma^2(Distance(T_u, C))$ are the **min**, **max**, **mean** and **variance** of the the distance between the locations in T_u vector and data collection center C for user u.*

Definition 22 *(User Country Count) For user u, $CountryCount(u) = |Country(T_u)|$ is the number of countries user u visited.*

Considering the features explained in Definitions 16 to 22 the produced multidimensional features tuple for user u is:

$$\begin{aligned} F_u = <& Y_u, argminEntropy(Y_u), argmaxEntropy(Y_u), \mu(Entropy(Y_u)), \\ & \sigma^2(Entropy(Y_u)), argminUserCount(Y_u), argmaxUserCount(Y_u), \\ & \mu(UserCount(Y_u)), \sigma^2(UserCount(Y_u)), argminFrequency(Y_u), \\ & argmaxFrequency(Y_u), \mu(Frequency(Y_u)), \sigma^2(Frequency(Y_u)), \\ & argminDistance(T_u, C), argmaxDistance(T_u, C), \mu(Distance(T_u, C)), \\ & \sigma^2(Distance(T_u, C)), LocationCount(u), CountryCount(u) > \end{aligned} \tag{5.6}$$

5.3.2 Feature Vectorization

The User Types Vector Y_u (Definition 16) has variable length which results in variable length tuple F_u for user u as the size of Y_u varies from one user to another. We employed the TF-IDF (Equation 5.9) vectorization technique to vectorize Y_u to a fixed length feature set for all users u in U. Although all items in Y_u are important, there are highly repeated location types in Y_u which carry less meaning than other types to cluster users. For instance, assuming that if location type *store* is repeated in 80% of the entire database D, considering this weight to find similarities does not provide reasonable results as it is believed that a highly frequent item does not necessarily indicate higher importance. Conversely, considering less frequent items have higher potential to obtain similarities of two input items. Therefore, similar to text-mining, this technique considers some *location types* as stop words or noise. To fulfill this consideration, TF-IDF reduces the effect of highly frequent types in Y_u.

$$tf_{i,j} = \begin{cases} 1 + log_2 f_{i,j} & if f_{i,j} > 0 \\ 0 & otherwise \end{cases} \tag{5.7}$$

$$idf_i = log\frac{N}{n_i} \tag{5.8}$$

$$tf_{i,j} - idf_i = tf_{i,j} \times idf_i \tag{5.9}$$

5.3.3 Feature Normalization

Because the features engineered by TF-IDF method are inherently sparse, it is believed that the direction of the vector provides more meaning than the magnitude of the vector [75]. As a result, the extracted features are normalized to become unit-length vectors. For instance, in a study, Zhong et al.[76] showed that unit-length TF-IDF vectors provide better clustering results than multinomial or multivariate Bernoulli models. To achieve this advantage, we employed *L2-norm* (Equation 5.10) to normalize TF-IDF results.

$$\begin{aligned} x_{norm} &= \frac{x}{\|x\|_2} \\ x_{norm} &= \frac{[x_1, x_2, ..., x_n]}{\sqrt{x_1^2 + x_2^2 + ... + x_n^2}} \end{aligned} \tag{5.10}$$

In addition to the location types, we apply the *l2-norm* on other 18 features provided in Equation 5.6.

The outcome of *TF-IDF vectorization* and *l-2 normalization* techniques is to compute fixed length feature vector for all users in U. Therefore, we present a new form for database D as $D_{refined}$:

$$\mathrm{D_{refined}} = \begin{bmatrix} x_{u_1}^1 & x_{u_1}^2 & \dots & x_{u_1}^n \\ x_{u_2}^1 & x_{u_2}^2 & \dots & x_{u_2}^n \\ \vdots & \vdots & \ddots & \vdots \\ x_{u_k}^1 & x_{u_k}^2 & \dots & x_{u_k}^n \end{bmatrix} \tag{5.11}$$

where for each row at database $D_{refined}$ the features have fixed length of size n and each data point $x_{u_j}^i$ contains normalized values for user j and feature i.

5.3.4 Clustering Users' Feature Vectors

We employed *k-Means* [34] clustering algorithm to group users with similar features. It is shown how spherical k-Means [75, 27] have reasonable performance in clustering high dimensional data particularly in vectorized and normalized form. The normalized data transforms the data space to spherical form that each value has unit-length distance from the center of the multidimensional space. To find the similarities between the normalized and unit-length vectors, cosine similarity has shown better

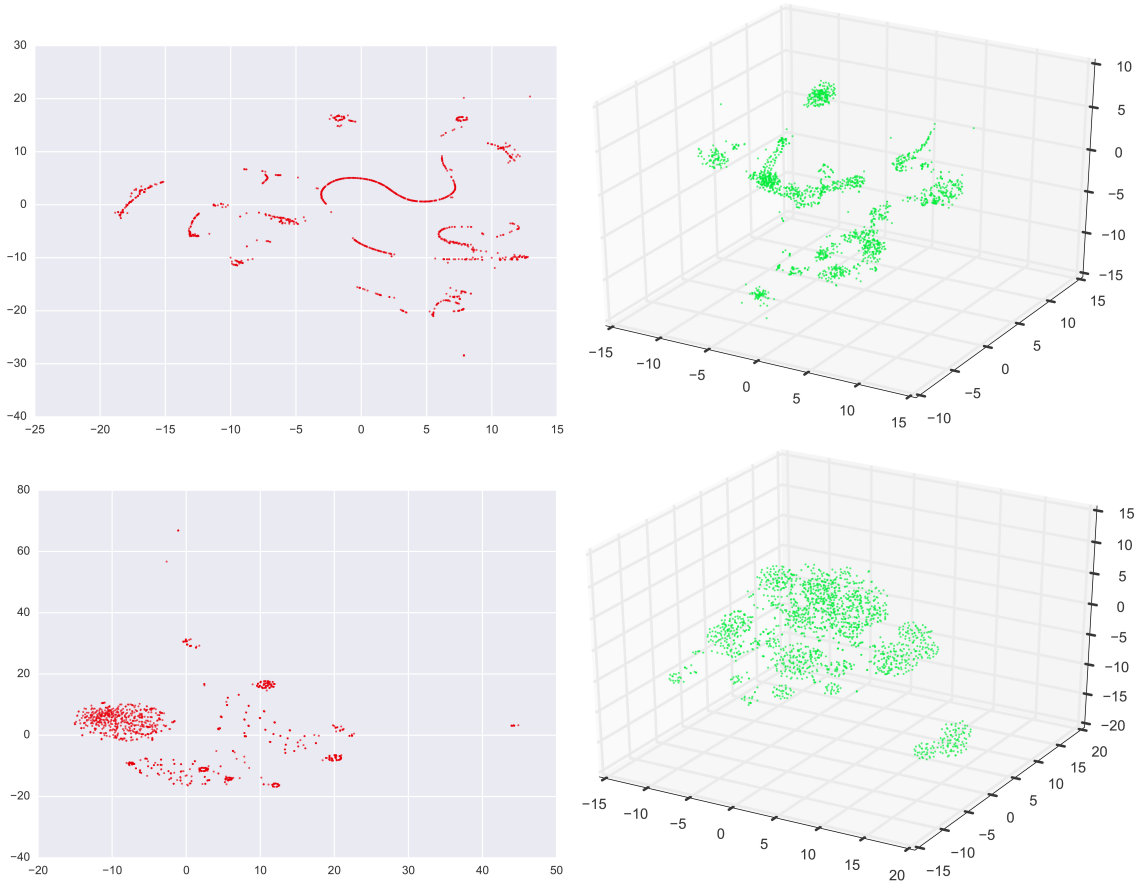

Figure 5.11: The representation of database $D_{refined}$ (Equation 5.11) before (top) and after (bottom) normalization process. The representation is done by applying t-SNE [65] dimensionality reduction technique to 2D (red) and 3D (green) space.

performance than Euclidean distance [63, 75]. The intuition behind it is that the direction of a unit-length vector is more important than the magnitude of such vector as a similarity metric. The cosine distance is defined by:

$$sim_{cos}(A, B) = cos(A, B) = \frac{\vec{A}.\vec{B}}{\|\vec{A}\| \, \|\vec{B}\|} \tag{5.12}$$

The $sim_{cos}(A, B)$ function measures the similarity of two vectors by comparing the *cosine of angle* between of the feature vectors A and B. The cosine value for the similarity metric is a value between -1 (highest dissimilarity) and 1 (highest similarity) that is used to measure the distance between two feature vectors.

The outcome of the k-Means algorithm is k clusters which their data points have minimum intra-cluster and maximum inter-cluster similarity. It assigns all n data

points x_{u_i} to exactly k clusters. The convergence criterion for k-Means algorithm is to obtain the least *sum of squared error* E for the members of their respective clusters *w.r.t* the centroid of their clusters (c_1, c_2, ..., c_m).

$$E = \sum_{i=1}^{k} \sum_{x_{u_i} \in c_p} sim(x_{u_i}, c_p)^2 \tag{5.13}$$

To discover the best number for k of k-Means algorithm, we used Silhouette analysis [60] by calculating the *Silhouette Coefficient* for each input item x_{u_i}. The intuition behind silhouette analysis is to find whether items are assigned confidently, marginally, or loosely to their assigned clusters. The average of the silhouette coefficient of the entire data points, shows the correctness of parameter k which the higher average of coefficients shows a better value for parameter k. This method is explained in detail in Subsection 6.1.

To sum up this chapter, the entire process of mapping SSID data to actual locations, feature engineering, vectorization, normalization and clustering is brought below:

1. Map SSID data points to actual locations using Google Places API.
2. Extract Types, Countries, and Coordinates of the locations from the results of step 1.
3. Calculate Entropy(Y_x), UserCount(Y_x), and Frequency(Y_x) for all types Y of location x.
4. Calculate the distance for location x from data collection point C as Distance(x, C).
5. Extract visited countries as Country(x).
6. Concatenate features generated for each location in user Partial Spatial History vector T_u for user u:
   (a) Calculate TF-IDF for User Types Vector Y_u.
   (b) Calculate $argmin$, $argmax$, μ, and σ^2 for UserCount(Y_u), Frequency(Y_u), and Entropy(Y_u) for user u.

(c) Generate $argmin$, $argmax$, μ, and σ^2 for $Distance(T_u, C)$, and $CountryCount(T_u)$ for all locations in vector T_u.

(d) Normalize TF-IDF vectors with $L2 - norm$.

(e) Normalize extracted features at step **6b** and **6c** using *L2-norm*.

(f) Concatenate the entire normalized features in new database $D_{refined}$.

7. Discover the best value for k using Silhouette Analysis.

8. Cluster the users into k clusters from database $D_{refined}$ using k-Means algorithm with Cosine distance function.

Chapter 6

Evaluation of SSID Clustering Task

In this chapter, we present the results of the clustering task on SSID data. At first, we explain our evaluation method. Then, we present and interpret the resulting clusters. Due to the lack of ground truth dataset in this task, instead of quantitatively evaluating the performance of the clustering task, we describe the amount of insights we could gain in a real-world scenario from the captured SSID data.

6.1 Evaluation Method

Collecting gold standard or ground truth dataset was not feasible in the scope of this master thesis. Therefore, we designed our evaluation method independent of ground truth dataset. In the lack of ground truth data, the performance evaluation of a clustering task is not trivial compared to a classification task where calculating a performance score like precision or AUC ROC would be adequate. Instead, we pursued a different path. In this method, we would like to achieve the most meaningful clustering model that 1) has reasonable performance scores independent of the ground truth data and 2) meaningfully separates the population based on the their spatial behaviours. In order to achieve this goal, first we propose our method on how to optimize the parameters of the clustering algorithm which is quantitatively reliable regardless of the ground truth data, then, we propose our approach to interpret the clusters to show how the optimized model performs on the $D_{refined}$ dataset.

Before explaining the evaluation techniques, we should mention the minimum acceptance criteria for a clustering task which is defined below:

Definition 23 *(Minimum Acceptance Rules) Let C be the result of a clustering task containing individual clusters and $C_j \in C$ be one of the clusters, the minimum acceptance criteria for a clustering task should satisfy the following two conditions:*

- $|C| > 1$. *In word, let $|C|$ be the number of the clusters in a clustering, the*

number of clusters should be more than one.

- $|C_j| > 1$. *In words, let* $|C_j|$ *be the size of cluster* C_j*, the number of elements assigned to this cluster should be more than 1.*

A clustering model with only one cluster does not convey any insight as the main objective of a clustering task is to separate the data into different groups of similar objects. In addition, a cluster in a clustering model with only one element does not provide any information as we would like to see how objects are similar to each other within a cluster.

6.1.1 Parameter Optimization

The most important parameter of *k-Means* algorithm is the number of clusters or parameter k. In order to find the best value for this parameter we employed Silhouette Analysis technique. The outcome of utilizing this method is to reach the most confident model in the lack of the ground truth data. Silhouette Coefficient $sil(x_{u_i})$ for input item x_{u_i} is calculated by Equation 6.3:

$$a(x_{u_i}) = \frac{1}{|C_A| - 1} \sum_{x_{u_i} \in C_A, x_{u_j} \neq x_{u_i}} sim(x_{u_i}, x_{u_j}) \tag{6.1}$$

$$b(x_{u_i}) = min_{C_B A} \frac{1}{|C_B|} \sum_{x_{u_i} \in C_B} sim(x_{u_i}, x_{u_j}) \tag{6.2}$$

$$sil(x_{u_i}) = \frac{b(x_{u_i}) - a(x_{u_i})}{max\{b(x_{u_i}),\, a(x_{u_i})\}} \tag{6.3}$$

where $a(x_{u_i})$ is the calculated distance between data point x_{u_i}, and its respective cluster C_A and $b(x_{u_i})$ is the distance of the data point from its nearest cluster C_B where $x_{u_i} \notin C_B$. The output of $sil(x_{u_i})$ function shows the confidence of assignment of data point x_{u_i} to cluster C_A. The resulting $sil(x_{u_i})$ coefficient has a range of $[-1, 1]$. Value close to $+1$ shows the sample is away from its neighbours. In other words, we can infer that it is correctly assigned to its cluster C_A. Value approaching to 0 shows that the data point is close to the decision boundary of two or more clusters. In other words, its assignment is not very confident. Finally, values close to -1 show that the data point is closer to a neighbouring cluster C_B than to its assigned cluster C_A which means its assignment should be wrong.

Silhouette Coefficient can be extended to evaluate the quality of individual clusters by computing the average of Silhouette Coefficients of items assigned to cluster C_j as brought in Equation 6.4. The quality of the entire clustering task can be viewed as the average of Silhouette Coefficients of all clusters defined in Equation 6.5.

$$sil(C_j) = \frac{1}{|C_j|} \sum_{x_{u_i} \in C_j} sil(x_{u_i}) \tag{6.4}$$

$$sil(C) = \frac{1}{k} \sum_{i=1}^{k} sil(C_i) \tag{6.5}$$

The provided Figures 6.1 and 6.2, show the Silhouette analysis results of applying clustering task for k = [2, 20, 40, 60, 80, 100] on database $D_{refined}$. The applied dimensionality reduction method is t-Distributed Stochastic Neighbor Embedding (t-SNE) that is explained in Section 6.2. The left plots show the Silhouette Coefficient for each cluster. Within each cluster, Silhouette Coefficients of individual data points are displayed and ordered from lowest (bottom) to highest (top). The red dashed vertical line shows the average of Silhouette Coefficients and the blue lines are the standard deviation of the coefficients. The right plot shows the results of applying k-Means on the dataset $D_{refined}$. The graph is plotted based on the projection of $D_{refined}$ dataset on a 2D surface for visualization purpose. Each cluster is displayed with a similar color in both left and right plots.

We ran the entire process for k = [2..200] on the $D_{refined}$ dataset to find the best parameter k based on the average and standard deviation of Silhouette Coefficient for all clustering tasks. The result of the of this experiment is provided in Figure 6.3. This figure shows the upward trend of average and standard deviation of Silhouette Coefficient as the number of clusters increases. The most significant growth is between 7 and 22 clusters that the average grows by almost 15 percent from 0.39 to 0.55. From $k = 23$ to $k = 37$ the growth is still sharp. After $k = 37$, the growth rate decreases. On the other hand, the standard deviation of Silhouette Coefficient shows no change after $k = 7$. As the number of clusters grows, the average Silhouette Coefficient value increases because of the fact that smaller clusters are more likely to contain more homogeneous data points than larger clusters. On the other hand, a big number for k reduces the readability of the clustering task. Therefore, there should be a balance

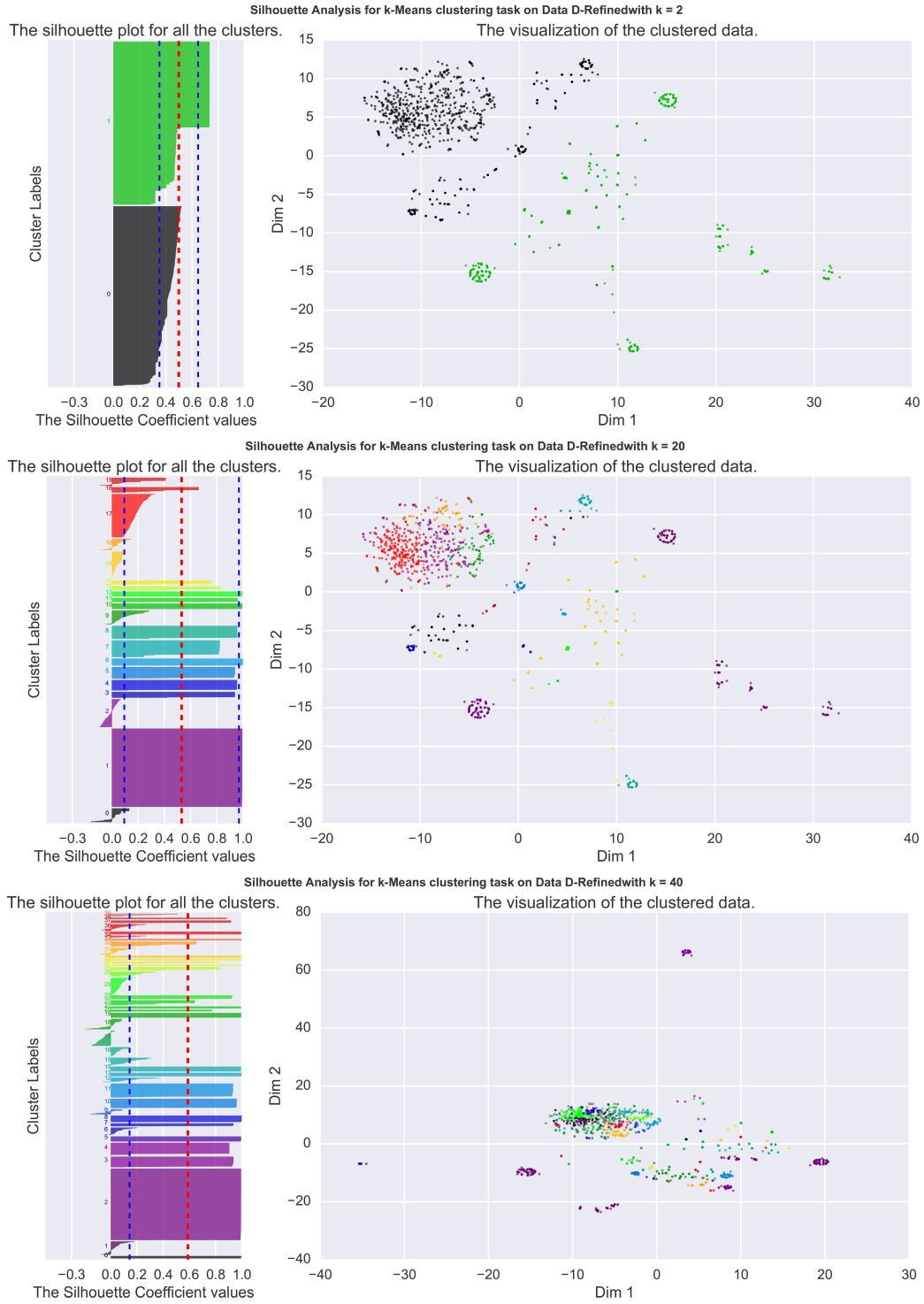

Figure 6.1: Silhouette Coefficient values for $k = 2$ (top), 20 (middle) and 40 (bottom).

between the Silhouette Coefficient and the number of clusters.

Considering provided discussion, we chose k = 37 for the number of clusters on

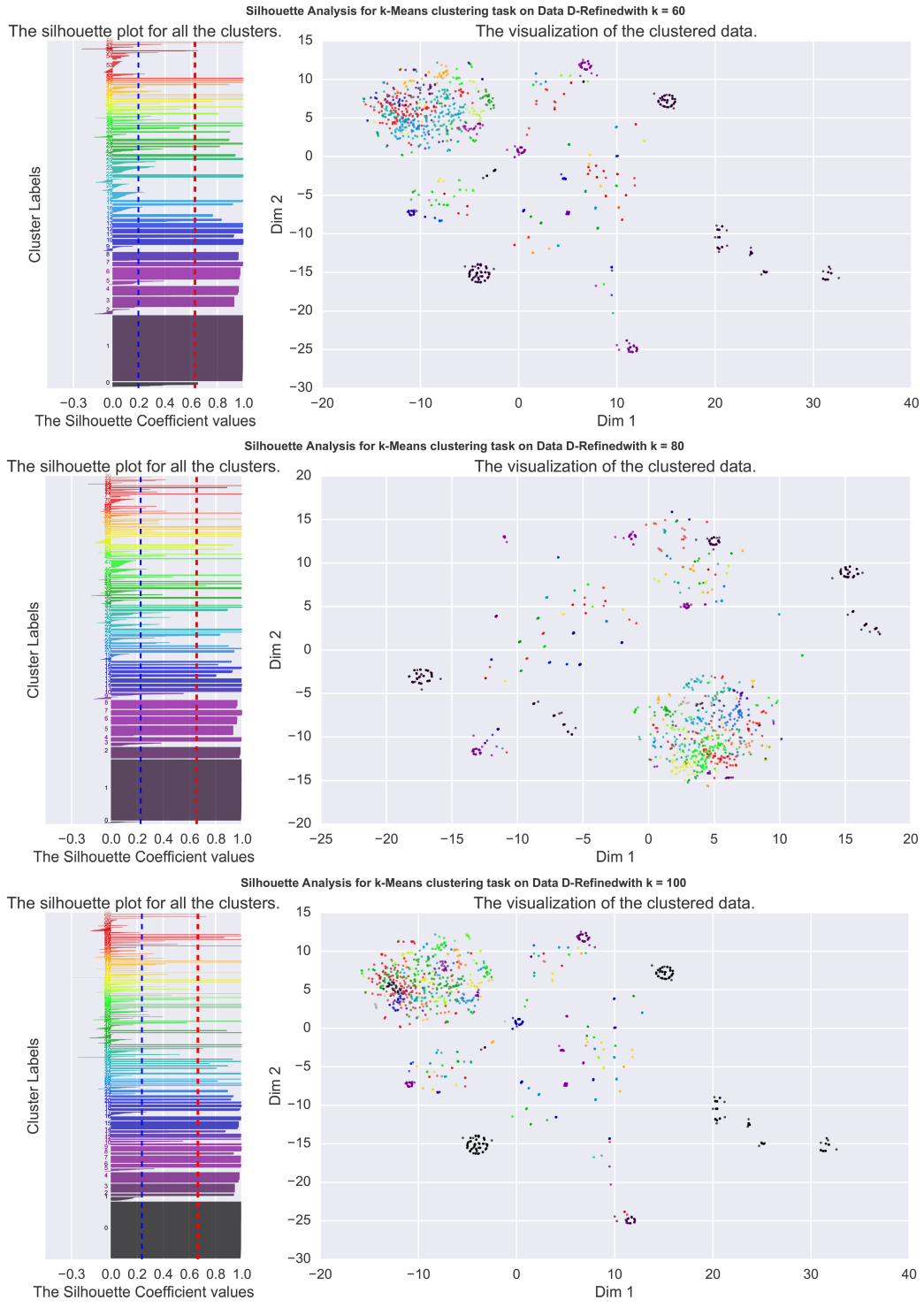

Figure 6.2: Silhouette Coefficient values for $k = 60$ (top), 80 (middle) and 100 (bottom).

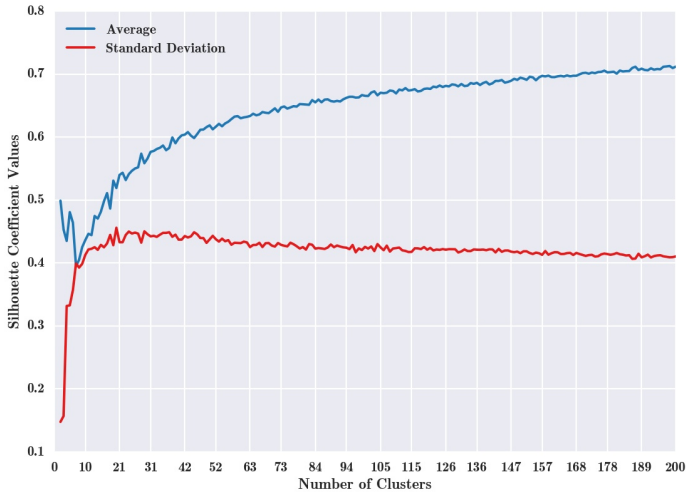

Figure 6.3: Silhouette Coefficient average and standard deviation values for $k = 10$ to 200.

the $D_{refined}$ dataset to maintain the balance between the Silhouette Coefficient and the interpretability of the clustering task. Moreover, we would like to satisfy the Minimum Acceptance Rules defined in Definition 23.

6.2 Visualizing Clustering Results

Visualizing high dimensional data is a non-trivial task. As humans, we can comprehend objects in maximum 3D space. In a 3D space, although we can visualize maximum of 10-15 features (e.g. size of bubble, color, shape, etc.), the visualization loses its comprehensibility. Therefore, instead of maximizing the number of possible plottable features in a 3D space, there is another way to visualize high dimensional data by utilizing dimensionality reduction techniques which have a similar objective. We employed a dimensionality reduction technique that reduces the number of features for all the data points whilst keeps the structure of the data points in the mapping process from high dimensional space to low dimensional space.

The employed algorithm is t-SNE[65] that maps the original high dimensional dataset $D_{refined}$ to a low dimensional dataset for visualization purpose. The original dataset $D_{refined}$ consists of 125 dimensions which 107 of the features correspond to

the User Types Vector (Definition 16) and the other 18 dimensions represent the engineered features for Entropy (Definition 17), User Count (Definition 18), Frequency (Definition 19), Distance (Definition 21), Country Count (Definition 22) and Location Count (Definition 20).

t-SNE dimensionality reduction technique tries to map a high dimensional dataset into n dimensional space (mostly 2D or 3D) using a variation of Stochastic Neighboring Embedding (SNE) [36] which arguably provides much better visualization results compared to other dimensionality reduction techniques [65] by preserving much of the local and global structure of high dimensional dataset in the mapping process. SNE that is the essence of t-SNE, converts the original dataset into a pairwise similarity matrix for all the data points. The similarity of data point x_i to x_j is the conditional probability $p(j|i)$ (Equation 6.7), which shows how probable x_i would choose x_j as its neighbour.

$$p(j|i) = \frac{e^{\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}}} \tag{6.6}$$

For the data points x_i and x_j in high dimensional space and y_i and y_j in low dimensional space, using Equation 6.6, the SNE produces the same conditional probability as $q(j|i)$ for y_i and y_j in low dimensional space.

$$q(j|i) = \frac{e^{-\|y_i - y_j\|^2}}{\sum_{k \neq i} e^{-\|y_i - y_k\|^2}} \tag{6.7}$$

Finally, the aim of t-SNE is to find the best values for y_i and y_j that minimizes Kullback-Leibler divergence (Equation 6.8) between joint distribution $p(i,j) = \frac{p(j|i)+p(i|j)}{2n}$ and joint distribution $q(i,j) = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1+\|y_k - y_l\|^2)^{-1}}$. In other words, it finds a mapping process acceptable when for data points y_i and y_j, conditional probability $q(j|i)$ is closest to $p(j|i)$.

$$\begin{aligned} C = KL(P\|Q) &= \sum_i \sum_j p(i,j) \log \frac{p(i,j)}{q(i,j)} \\ &= \sum_i \sum_j p(i,j) \log p(i,j) - p(i,j) \log q(i,j) \end{aligned} \tag{6.8}$$

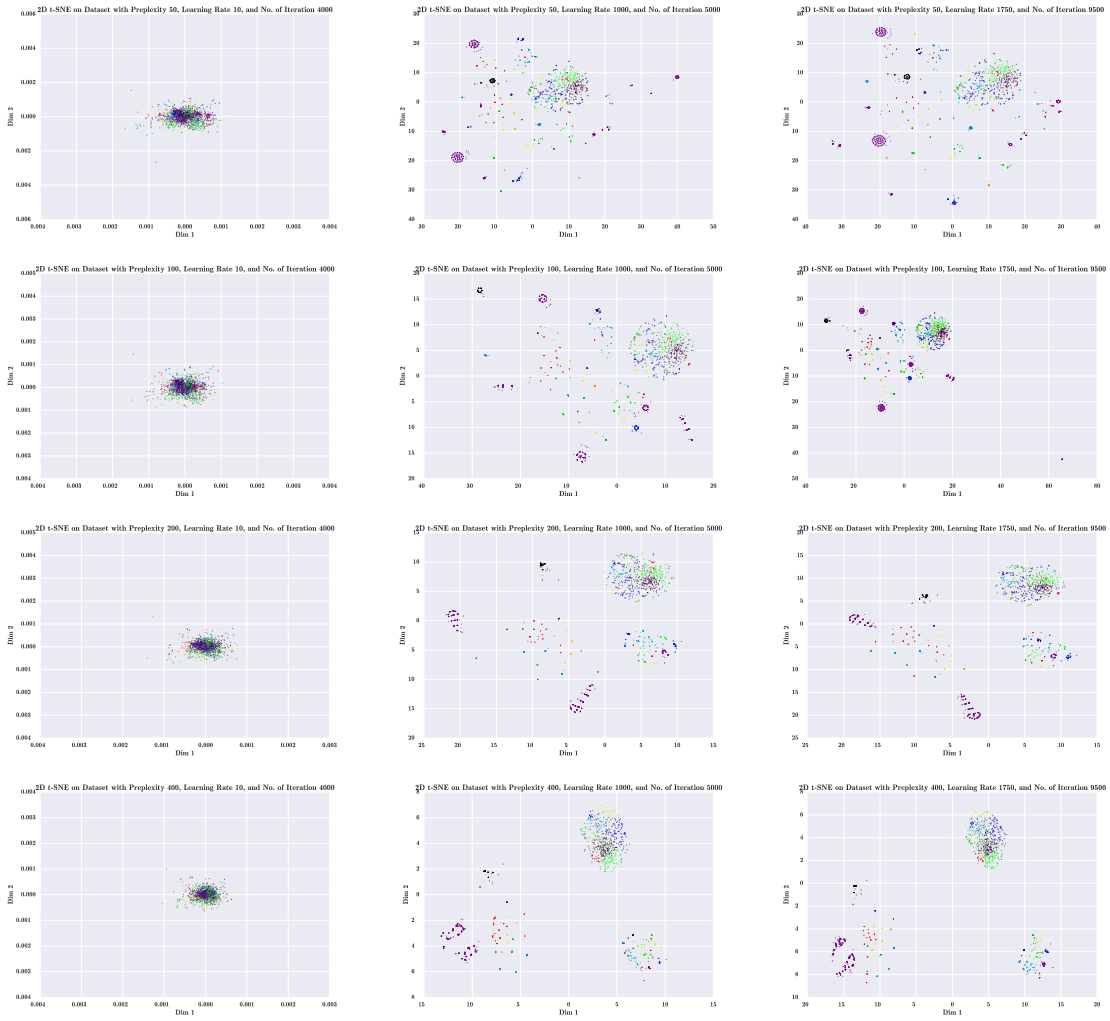

Figure 6.4: The results of the t-SNE hyper parameter tuning. The graphs are plotted by different values for each parameter: Perplexity = [50, 100, 200, 400], Learning Rate = [10, 1000, 1750], and No. of Iterations = [4000, 5000, 9500]

Amongst several hyper parameters to optimize t-SNE, there are three hyper parameters that their different values produce significantly different mapping results. These hyper parameters are *Perplexity*, *Number of Iterations*, and *Learning Rate*. To the best knowledge of the author, there is no proposed method to quantitatively tune the hyper parameters of the t-SNE. Therefore, we pursued a greedy approach, in which, given a wide range of values for these hyper parameters, we tried to find a set of parameters that best represent the clusters from visual stand point. Some of the results of this experiment are provided in Figure 6.4.

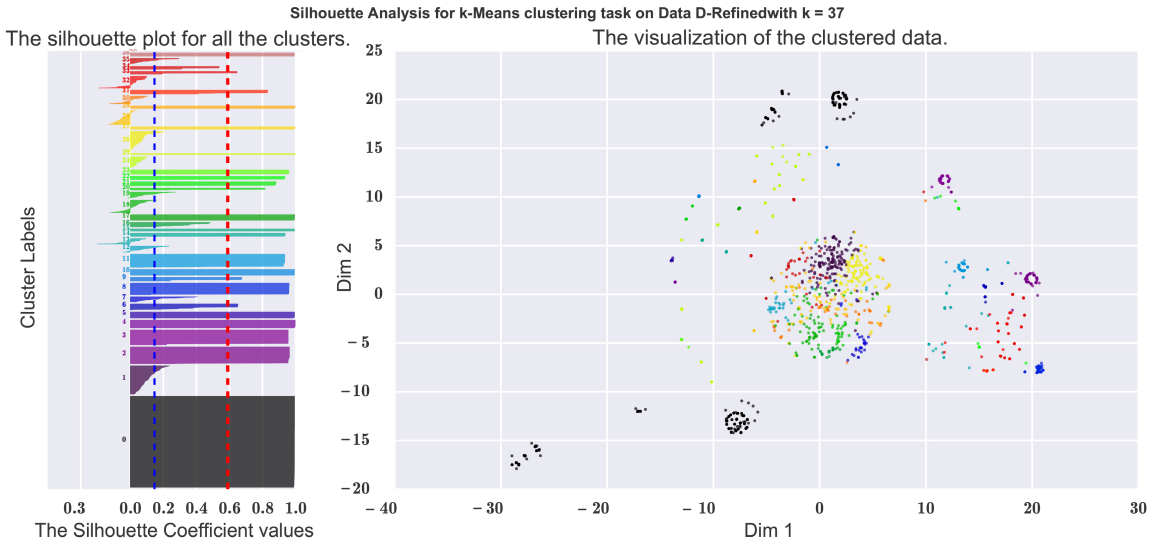

Figure 6.5: The final result of applying k-Means clustering algorithm on $D_{refined}$ after the Silhouette Analysis and t-SNE hyper parameter tuning.

6.3 Clustering Results

As we chose $k = 37$ for the number of clusters, k-Means algorithm assigns all the data points to 37 distinct clusters. The 2D and 3D representation of this task are illustrated in Figure 6.5. The clustering task interestingly separates individuals based on the extracted spatial behavioural features. In this section, the final outcome of the clustering task is explained.

The consequence of collecting one SSID for the majority of the understudied population, is producing a sparse matrix that consists of overwhelmingly zero entries for the User Types Vector. In addition, there are few types that considerably appeared in the majority of User Types Vectors, this causes the majority of entries have very similar feature vectors. For these group of users, the k-Means algorithm can group them into clusters with high Silhouette Coefficients. For example, cluster number 0 which contains the largest number of data points consists of users who just visited locations with location type *Crop*. Consequently, it achieved 1 for Silhouette Coefficient. On the other hand, cluster number 28 has 16 members whose feature values are completely diverse. Therefore the Silhouette Coefficient for cluster C_{28} is -0.1. The weight of User Types Vector Y_u is more considerable compared to other 18

features because of the larger number of members (107) and overwhelmingly zero values. Therefore, clusters with similar Y_u but diverse values for the rest of the features, are clustered confidently and obtain high Silhouette Coefficient. On the other hand, clusters with diverse Y_u but similar values for the rest of the features are clustered less confidently and obtain a Silhouette Coefficient value close to zero. The lowest Silhouette Coefficient value (close to -1) is assigned to clusters with completely diverse values for the entire feature vector F_u. The much higher effect of User Types Vector Y_u us intentional as we believe it has more importance in clustering people with similar spatial behaviours compared to the rest of the features in F_u.

We gathered the center of the clusters and listed in Table 6.1. In order to improve the readability of the table for four features of *Entropy*, *UserCount*, *Frequency* and *Distance*, we just brought the *mean* (μ) attribute in the table. For the same reason, we categorized the numerical values such as entropy into three groups of *Low*, *Med*, and *High*. Considering the range of values for each attribute, the categorization process of the numerical features follows the following rules:

- Entropy:
  - Low: $\mu < 5$
  - Med: $5 \leq \mu < 9$
  - High: $9 \leq \mu$
- Frequency:
  - Low: $\mu < 270$
  - Med: $270 \leq \mu < 840$
  - High: $840 \leq \mu$
- User Count:
  - Low: $\mu < 200$
  - Med: $200 \leq \mu < 450$
  - High: $450 \leq \mu$
- Distance:

  - Low: $\mu < 2,500,000$
  - Med: $2,500,000 \leq \mu < 8,500,000$
  - High: $8,500,000 \leq \mu$

- Location Count:
  - Low: $LC < 5$
  - Med: $5 \leq LC < 10$
  - High: $10 \leq LC$

- Country Count:
  - Low: $CC < 2$
  - Med: $2 \leq CC < 3$
  - High: $3 \leq CC$

Overview of each cluster is provided in Table 6.1. As explained above, the numerical features are categorized to increase the readability of the results.

The first column, *No.* is the cluster number which starts with 0. *SC* is the Silhouette Coefficient of the cluster C_j, the third column *Size* shows the number of elements in cluster C_j, *UTY* corresponds to the *User Types Vector* Y_u, where u is the user whose feature vector is at the center of cluster C_j, *Ent* is the *mean* (μ) of Entropy for user u, *UC* is the *mean* (μ) of User Count for user u, *Freq.* is the *mean* (μ) of Frequency for user u, *Dist.* is the *mean* (μ) of Distance feature for user u, *CC* is the Country Count for user u, and *LC* is the Location Count for user u. If the value in *UTY* is a maximum of three-length tuple, we listed the whole elements, but if it contains more than three elements, we indicated with keyword *Diverse* but is explained later. Below, we explain each cluster to discuss how the population is clustered by describing the members and their values. For some of the clusters, we also provided our hypothesized description of the potential type of the people that are members of a cluster.

[1]The table headers are as follows: No. → Cluster Number, SC → Silhouette Coefficient, UTV → User Types Vector, Ent → Entropy, UC → User Count, Freq. → Frequency, Dist. → Distance, CC → Country Count, LC → Location Count

Table 6.1: The values of the 37 cluster centers

No.	SC	Size	UTV	Ent	UC	Freq.	Dist.	CC	LC[1]
0	1	473	Corp.	High	High	High	Low	Mid	Low
1	0.1	148	Diverse	Low	Low	Low	Low	Mid	High
2	0.9	31	Bar, Restaurant	High	High	High	Low	Mid	Low
3	0.9	69	Store	High	High	High	Low	Low	Low
4	1	32	Bar	Low	Low	Low	Low	Mid	High
5	1	29	University	Low	Low	Low	Low	Mid	Low
6	0.7	18	Cafe, Bar, Restaurant	Mid	Low	Low	Low	Low	Low
7	0.25	12	Diverse	Mid	Low	Low	Low	Low	Low
8	0.9	24	Electronics Store	Mid	Low	Low	Low	Mid	Mid
9	0.35	30	Beauty, Hair Salon	Mid	Mid	Mid	Low	Low	Low
10	1	65	Store	Low	Low	Low	Low	Mid	Mid
11	0.9	32	Restaurant	Low	Low	Low	Low	Mid	High
12	0.2	88	Diverse	Mid	Mid	Mid	Mid	Mid	Low
13	-0.05	11	Diverse	High	High	High	Low	Mid	Low
14	0.9	23	Airport	High	High	High	Low	Mid	Low
15	1	18	School	Mid	Low	Low	Low	Low	Low
16	0.3	23	Travel Related	Low	Low	Low	Low	Mid	Low
17	1	58	Lodging	Low	Low	Low	Low	Low	Low
18	0	7	Travel Related	Low	Low	Low	Low	Low	Low
19	0.1	105	Diverse	Low	Low	Low	Low	Mid	High
20	0.9	14	Food	Mid	Mid	Mid	Low	Low	Low
21	0.9	74	Corp.	Low	Low	Low	Low	Mid	Mid
22	0.9	14	Clothing Store	Low	Low	Low	Low	Low	Low
23	0.9	78	Real Estate Agency	High	High	High	Low	Mid	Low
24	0.1	41	Diverse	Low	Low	Low	Low	Mid	Low
25	1	21	Insurance Agency	Low	Low	Low	Low	Low	Low
26	0.2	62	Diverse	Low	Low	Low	Low	Low	Low
27	1	15	Health	Low	Low	Low	Low	Low	Low
28	-0.1	16	Diverse	Mid	Low	Low	Low	Low	Low
29	1	32	Finance	Low	Low	Low	Low	Mid	High
30	0	18	Diverse	Low	Low	Low	Low	Mid	Low
31	0.8	41	Furniture Store	Mid	Mid	Mid	Low	Low	Low
32	0	32	Diverse	Mid	Low	Low	Low	Mid	Low
33	0.7	32	Night Club	Mid	Low	Low	Low	Low	Low
34	0.5	27	Hospital, Church	Mid	Low	Low	Low	Mid	Mid
35	0.2	61	Diverse	Mid	Low	Low	Mid	Mid	Low
36	1	20	Doctor	Low	Low	Low	Low	Low	Low

- The 1st cluster which contains the largest number members with highest possible Silhouette Coefficient, contains the users who visited one location with User

Types Vector (Y_u) *Corp*. As these users have very similar Y_u, the rest of their features are very similar. From this cluster we can possibly extract the name of their companies they visited or probably they work(ed). People in this cluster have visited 2-3 countries. We can call this cluster, the cluster of professionals.

- The 2$^{\text{nd}}$ cluster which contains the second largest population, consists of items which have diverse Y_u feature vectors but very similar values for the rest of the features. These people have visited mostly food and beverage (F&B) related locations. In general, they visited location types with homogeneous population and visited a large number of locations mostly in the city of Halifax. They also visited at least 2-3 foreign lands. This cluster may contain people with good socioeconomic status who enjoy spending time in F&B related locations.

- The 3$^{\text{rd}}$ cluster contains users who visited several locations mostly bars and restaurants located in Halifax. The members of this cluster are very similar to the members of the 2$^{\text{nd}}$ cluster.

- The 4$^{\text{th}}$ cluster contains users who visited locations only with type *Store* and have less significant diverse values for the non-type features (features other than Y_u). In general, they visited few locations and travelled only in Halifax. It is possible that the members of this cluster are sales assistants of retail stores.

- The 5$^{\text{th}}$ cluster contains users with completely similar feature vector F_u. They visited several locations with type *Bar*. The members of this cluster and the 2$^{\text{nd}}$ cluster have high similarities.

- The 6$^{\text{th}}$ cluster contains users who visited one or more universities in the past/present and visited 2-3 countries. The feature vector F_u of these people is very similar. This cluster may contain international students.

- The 7$^{\text{th}}$ cluster contains users with similar feature vector F_u. They mostly visited *cafes*, *bars* and *restaurants* only in Halifax.

- The 8$^{\text{th}}$ location contains users with similar non-type features but diverse User Types Vector Y_u. They visited several locations with less frequent location types in only Halifax.

- The 9^{th} cluster contains users who visited several electronic stores in 2-3 countries. People in this cluster are very similar to population of 8^{th} cluster.
- The 10^{th} cluster contains users who visited beauty and hair salons with diverse values for non-location type vectors. In general, they visited 1-3 locations in Halifax. They may probably work in salons.
- The 11^{th} cluster is similar to the third cluster which contains users who visited a store. However, the users in this cluster visited much more locations with higher frequent types.
- The 12^{th} cluster contains users who visited several restaurants in 1-2 countries.
- The 13^{th} cluster contains users who visited diverse locations with similar non-type features (features other than Y_u in F_u). They mostly visited health and beauty related locations. They also travelled to several locations in 2-3 countries.
- The 14^{th} cluster contains users who have diverse values for the entire feature vector F_u. In general, they visited few health and sports related locations in 2-3 countries.
- The 15^{th} cluster contains users who have been to only airports mostly in 2-3 countries. This cluster may contain people who are tourist.
- The 16^{th} cluster contains users who visited only few schools in Halifax. This cluster may contains local students.
- The 17^{th} cluster contains users who visited several travel and tourist related locations in 4-5 countries. This cluster may contain people who are tourist or active travellers.
- The 18^{th} cluster contains users who visited few hotels or resorts with completely similar feature vector F_u. Similar to the 15^{th} and 17^{th} clusters, this cluster may contain people who are tourist or active travellers.
- The 19^{th} cluster contains users who visited few diverse locations. In general, the locations are travel related and in 1 country.

- The 20th cluster contains users who visited many locations in 1-2 countries. In general, their travelled locations are a combination of F&B and touristic locations.

- The 21st cluster contains users with similar spatial history. They visited few food related locations in 1 country. They should be food lovers.

- The 22nd cluster contains users who visited several corporations in 1-2 countries. They can be professionals.

- The 23rd cluster contains users with similar Partial Spatial History. They visited few locations that were clothing store related in 2-3 countries.

- The 24th cluster contains similar elements. The users in this cluster visited few real estate agencies in 2-3 countries. They can be staff members of real state agencies.

- The 25th cluster contains users with diverse spatial history. They visited few locations in 1-2 countries. In general, the locations are entertainment related.

- The 26th cluster contains users with completely similar spatial history. They visited insurance agencies in the past. The members of this cluster can be people who work in insurance agencies or frequently visit one.

- The 27th cluster contains users with diverse spatial history but similar non-type feature vector. They visited several locations that are mostly food and health related.

- The 28th cluster contains users with completely similar spatial history. They visited few health related locations in Halifax.

- The 29th cluster contains diverse elements. The users in this cluster visited a large number of locations. Ignoring the most common location type which is *Corp.* the rest of location types are university or F&B related.

- The 30th cluster contains users who visited several locations with type *Finance* (e.g. banks or financial institutes).

- The 31st cluster contains users who visited few but diverse locations in the past. In general, they visited electronic related stores.

- The 32nd cluster contains users who visited few furniture stores in Halifax.

- The 33rd cluster contains diverse elements who visited few locations in 1-2 countries. The visited locations are generally related to services such as car repair shops and F&B related locations.

- The 34th cluster contains users visited few night clubs in Halifax.

- The 35th cluster contains users who visited few hospitals and religious (e.g. church, mosque) related locations.

- The 36th cluster contains users with diverse spatial history. In general, they visited health and business related (e.g. corporation) locations. They mostly visited several locations in 2-3 countries.

- The 37th and last cluster contains users who visited few health related locations in Halifax. This cluster may contain patients or health related professionals (e.g. doctors).

6.3.1 Discussion

The 37 clusters provided above show the possibility of clustering a population based on their SSID data which was non-intrusively collected. From the clustering task, the different types of users in a population based on their spatial behavioural history is distinguished. For example, the clustering result on the understudied population shows that the majority of the people are those who have been to a business unit (**probably** adult). Moreover, several of them visited several food and tourist related locations. In addition, the Spatial History shows that most of the population visited a foreign land (other than Canada) which *can be* an indication that they have strong socioeconomic status. A new unseen user can be clustered into one of these clusters based on the retrieved Partial Spatial History. Obtaining this knowledge from a population comes with no cost.

This framework is an example of what a brick-and-mortar business can achieve by clustering their customers based on their past visited locations. This data can provide invaluable insights about their customers to provide several business opportunities. For example, it helps them to directly target group of individuals with specific interests for marketing. Collecting such data from a neighbourhood can help advertisement agencies to provide related materials for the residents.

Combining this method with the proposed positioning method explained in Chapter 3 provides deeper insights into the understudied population as inside and outside population. For example, a venue owner can be empowered as she can study the characteristics of the inside customers and outside people separately to design different marketing strategies for each distinct population.

Chapter 7

Conclusion

In this thesis we presented two frameworks that each can extend the potentials of Wi-Fi Analytics platforms. The first contribution of this thesis was the proposal of a fingerprinting positioning system based on a new crowdsourcing method. The advantage of this method is its non-intrusiveness and low cost of implementation. We also showed how the implemented positioning system can be transferred to other locations in the lack of positioning data.

Second, we proposed a framework to cluster a population based on their spatial history. The advantage of this framework is its strength in extracting the spatial history in the first visit of a device. In addition, the framework is capable of understanding a population based on the extracted information. This framework extends the knowledge of brick-and-mortar businesses on their customers with a fast and inexpensive technique.

7.1 Positioning System

Our proposed fingerprinting positioning system is enriched with a new crowdsourcing technique that non-intrusively collects the fingerprints from crowds. We studied two sources of data as Registered (Definition 5) and Night (Definition 6) Datasets. First, using statistical and machine learning approaches we proved that the features extracted from these datasets are very similar to the features extracted from inside and outside populations.

In order to prove it, we first gathered 200 samples as ground truth datasets for inside (*SurveyDataI*) and outside (*SurveyDataO*) populations. Then, we visualized the dataset using three scatter plots to observe the visual and statistical distribution of the extracted features of the datasets. From the visualization we found that the Registered and SurveyDataI datasets have similar characteristics. In addition, we reached a similar conclusion for the Night and SurveyDataO datasets.

Then, we employed Andrew Darling (AD) test to validate the statistical similarity of Registered Dataset with the SurveyDataI dataset. We employed AD to perform the same experiment on Night and SurveyDataO Datasets. The AD test showed that, statistically, the Registered Dataset is very similar to the SurveyDataI and the Night Dataset is very similar to the SurveyDataO dataset.

The final experiment to test the similarity of these datasets was using machine learning techniques. We built two One-Class SVM and one Two-Class Random Forests classifiers to investigate the capability of the classifiers trained on Registered and/or Night Datasets in classifying the ground truth datasets. The performance of the classification tasks showed that the ground truth dataset is highly classifiable by the classifiers trained on Registered and/or Night Datasets.

Then we proposed the positioning system that was built on the investigated Registered and/or Night datasets to classify devices as inside or outside of a store. Through three different experiments, we achieved reasonable performance in estimating the device positions.

Finally, we showed that the positioning system build by Registered and Night datasets are generally transferable from one location to another in case the Registered and/or Night datasets are are not available there.

7.1.1 Future Work

One of the possible improvement of our proposed positioning system is to improve the performance of the classifiers. The current positioning system showed that in some situations, like imbalanced data and complex indoor environments, its performance drops considerably. One possible solution is applying windowing techniques to improve the quality of the classifiers in location estimation [25]. In addition, applying more complex signal-based feature extraction techniques such as extracting frequency or time-frequency domain features can potentially improve the positioning performance. Some immediate techniques can be Fourier transform for frequency domain features and Wavelet transform for time-frequency domain features [26]. These changes can potentially improve the performance of the positioning system in transfer learning task too.

Another possible future direction for the proposed positioning system is to identify

fine-grained positions. The current system is capable of identifying the coarse-grained position as whether a device is inside or outside of a location and the precise location information of the device is unknown. This is particularly due to the nature of the proposed method because it considers the whole RSSI vector as a unit in the prediction task. Windowing the Registered and Nights datasets into smaller chunks combined with a distance calculation function can potentially help to achieve this goal.

7.2 Clustering Partial Spatial History

The proposed clustering task to distinguish the type of a population based on their spatial behavioural features extracted from the Partial Spatial History dataset. The Partial Spatial History dataset is the dataset that contains the SSID records non-intrusively collected from the Preferred Network List (PNL) of Wi-Fi enabled devices.

Using Google Places API, we proposed our method to map the SSIDs to actual locations. In the mapping process we managed to extract several semantic features about the locations using Google Places API including *location name*, *location types*, and *location address*. Then, we applied data cleansing on entries which have no or negative contribution in the clustering task such as residential locations.

From these attributes, we extracted 125 features to describe the spatial behaviours of a user. First, we calculated the *Entropy*, *User Count*, and *Frequency* of each location type. Then, considering the total visited locations, we calculated the number of visited locations and countries as *Location Count* and *Country Count*. Finally, we calculated the distance between the data collection center (Goldberg Building and SolutionInc office) to measure the Distance feature for each location. Then, using TF-IDF technique we vectorized the types of the visited locations. This task produced a matrix with 107 attributes corresponding to 107 locations types at User Types Vector (Y_u) for user u. In order to vectorize the other extracted features we calculated the mean, variance, minimum and maximum of Entropy, Frequency, and User Count features for the types of the visited locations. We also calculated these four features for the Distance feature of the visited locations. Then by combining the Y_u and other features, we generated the user feature vector F_u for the each user u. Combining the entire feature vectors F for all users, the database D is generated.

We employed *l-2 normalization* on the entire database D to transform the vectors

into unit-length vectors. The resulting dataset is named dataset $D_{refined}$.

The refined dataset $D_{refined}$ is then clustered using k-Means clustering algorithm equipped with cosine-similarity function. The k for k-Means algorithm was discovered using Silhouette Analysis for the entire clustering task. Then, using t-SNE, the resulting clustering task is visualized. Finally, we described the result of the clustering task by explained the members of each cluster.

The resulting clusters shows that the extracted SSID data (Partial Spatial History) contains valuable insights about a the spatial behaviours of a population. It also shows that a brick-and-mortar business can leverage this freely accessible data and inexpensive method to extend their knowledge about their customers.

7.2.1 Future Work

Our proposed clustering method is capable of accurately clustering people with highly similar User Types Vectors. However, if the User Types Vector of some users contain long and diverse location types, the clustering algorithm is not able to provide a clear distinction among them. Therefore, people with long and diverse locations types are mostly grouped into one-two clusters. This makes interpreting of those clusters hard. An immediate solution for this problem is to reduce the location types vector by replacing similar types with a more general type. For example, bar, restaurant, and café are related to food and beverages (F&B). Therefore, we can replace these types with F&B. Such measures can reduce the size of User Types Vector significantly. Consequently, the number of features for the User Types Vector can be reduced to 7-8 instead of current 107 attributes as we can group the entire location types into 7-8 major groups. In other words, it helps to reduce the sparsity of the produced matrix as the TD-IDF vectorizer produces smaller matrix instead of a large one with overwhelmingly zero values.

The proposed method does not consider the semantic meaning of location types in the clustering task. For example, it does not consider the semantic similarity of location type *bar* and *restaurant*. A possible solution for this limitation is to use word embedding techniques such as Word2Vec [50]. In other words, by constructing a model that is capable of discovering semantic similarities among words, instead of directly using the location types, we can cluster location types based on the embedded

representation which contains the semantic information. Therefore, the clustering task is able to consider the similarities. For example, because the representation of *restaurant* and *bar* is very similar compared to *insurance agency*, the clustering algorithm considers this similarity in clustering the location types.

We believe this is just the tip of the iceberg. In addition to the technical improvements, we believe that the possible opportunities that SSID data can provide for several types of use cases is massive. However, we should consider the privacy threats in such improvements to make useful yet secure and safe systems.

7.3 Discussion

Wi-Fi Analytics platforms have proven their importance in a wide range of industries. However, their enormous applications does not neutralize the concerns over the potential breach of privacy. A possible improvement for both methods that are proposed in this thesis is to leverage privacy preserving techniques to avoid similar misuses. We have worked with the *Institut de Recherche en Informatique et Systmes Alatoires* and the *Institut National des Sciences Appliques de Lyon* in France to address the privacy concerns by leveraging methods that are designed for Wi-Fi data to preserve the privacy of individuals [8].

Bibliography

[1] `http://www.euclidanalytics.com/`. Accessed: 2016-07-24.

[2] `http://www.aislelabs.com/`. Accessed: 2016-07-24.

[3] `https://getturnstyle.com/`. Accessed: 2016-07-24.

[4] `http://www.purple.ai/`. Accessed: 2016-07-24.

[5] 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. *IEEE Std 802.11-1997*, pages i–445, 1997.

[6] AirSage. Pricing. [Online; accessed November 28, 2016].

[7] Abdullah Saad Mohammed Al-Ahmadi, Abdusamea IA Omer, Muhammad Ramlee Kamarudin, and Tharek Abdul Rahman. Multi-floor indoor positioning system using bayesian graphical models. *Progress In Electromagnetics Research B*, 25:241–259, 2010.

[8] Mohammad Alaggan, Mathieu Cunche, and Marine Minier. Privacy-Preserving t-Incidence for WiFi-based Mobility Analytics. In *7e Atelier sur la Protection de la Vie Privée (APVP'16)*, Toulouse, France, July 2016.

[9] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 2, pages 775–784 vol.2, 2000.

[10] Marco V Barbera, Alessandro Epasto, Alessandro Mei, Vasile C Perta, and Julinda Stefa. Signals from the crowd: uncovering social relationships through smartphone probes. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 265–276. ACM, 2013.

[11] Roberto Battiti, Nhat Thang Le, and Alessandro Villani. Location-aware computing: a neural network model for determining location in wireless lans. 2002.

[12] E. S. Bhasker, S. W. Brown, and W. G. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. In *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*, pages 111–120, March 2004.

[13] Philipp Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 55–60, New York, NY, USA, 2008. ACM.

[14] Bram Bonné, Peter Quax, and Wim Lamotte. Your mobile phone is a traitor!–raising awareness on ubiquitous privacy issues with sasquatch. 2014.

[15] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[16] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[17] Solomon Chan and Gunho Sohn. Indoor localization using wi-fi based fingerprinting and trilateration techiques for lbs applications. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38:4, 2012.

[18] Ningning Cheng, Prasant Mohapatra, Mathieu Cunche, Mohamed Ali Kaafar, Roksana Boreli, and Srikanth Krishnamurthy. Inferring user relationship from hidden information in wlans. Orlando, FL, 2012.

[19] M. Chernyshev, C. Valli, and P. Hannay. On 802.11 access point locatability and named entity recognition in service set identifiers. *IEEE Transactions on Information Forensics and Security*, 11(3):584–593, March 2016.

[20] Maxim Chernyshev, Craig Valli, Peter Hannay, undefined, undefined, undefined, and undefined. Service set identifier geolocation for forensic purposes: Opportunities and challenges. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 00(undefined):5487–5496, 2016.

[21] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.

[22] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 20152020 white paper, 2016. [Online; accessed November 11, 2016].

[23] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM, 2010.

[24] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. Linking wireless devices using information contained in wi-fi probe requests. *Pervasive and Mobile Computing*, 11:56–69, 2014.

[25] Vinita Daiya, Jemimah Ebenezer, SAV Satya Murty, and Baldev Raj. Experimental analysis of rssi for distance and position estimation. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 1093–1098. IEEE, 2011.

[26] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, Sep 1990.

[27] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[28] C. Drane, M. Macnaughtan, and C. Scott. Positioning gsm telephones. *IEEE Communications Magazine*, 36(4):46–54, 59, Apr 1998.

[29] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(11):1689–1689, 2009.

[30] Nathan Eagle, Alex (Sandy) Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.

[31] S. H. Fang, C. H. Wang, T. Y. Huang, C. H. Yang, and Y. S. Chen. An enhanced zigbee indoor positioning system with an ensemble approach. *IEEE Communications Letters*, 16(4):564–567, April 2012.

[32] Shih-Hau Fang and Tsungnan Lin. Principal component localization in indoor wlan environments. *IEEE Transactions on Mobile Computing*, 11(1):100–110, 2012.

[33] Google. Improve your local ranking on google, 2016.

[34] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[35] S. He and S. H. G. Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys Tutorials*, 18(1):466–490, Firstquarter 2016.

[36] Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 857–864. MIT Press, 2003.

[37] Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society.

[38] M. Kanaan and K. Pahlavan. A comparison of wireless geolocation algorithms in the indoor environment. In *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, volume 1, pages 177–182 Vol.1, March 2004.

[39] Shahrzad Khodayari, Mina Maleki, and Elham Hamedi. A rss-based fingerprinting method for positioning based on historical data. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*, pages 306–310. IEEE, 2010.

[40] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[41] Petri Kontkanen, Petri Myllymäki, Teemu Roos, Henry Tirri, Kimmo Valtonen, and Hannes Wettig. Topics in probabilistic location estimation in wireless networks. In *PIMRC*, pages 1052–1056, 2004.

[42] Silvio Lattanzi and D. Sivakumar. Affiliation networks. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 427–434, 2009.

[43] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD Inter. Conf. on Management of Data*, SIGMOD '07, pages 593–604, New York, NY, USA, 2007. ACM.

[44] Minkyu Lee, Hyunil Yang, Dongsoo Han, and Chansu Yu. Crowdsourced radiomap for room-level place recognition in urban environment. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 648–653. IEEE, 2010.

[45] Xinrong Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila. Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems. In *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152)*, volume 6, pages 3015–3020 vol.6, 2000.

[46] H. Lim, L. C. Kung, J. C. Hou, and H. Luo. Zero-configuration, robust indoor localization: Theory and experimentation. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–12, April 2006.

[47] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Trans. Sys. Man Cyber Part C*, 37(6):1067–1080, November 2007.

[48] A. Di Luzio, A. Mei, and J. Stefa. Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[49] J. Manweiler, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Predicting length of stay at wifi hotspots. In *INFOCOM, 2013 Proceedings IEEE*, pages 3102–3110, 2013.

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[51] Jeffrey Pang and Srinivasan Seshan. Tryst: The case for confidential service discovery. In *In HotNets VI: The Sixth Workshop on Hot Topics in Networks*, 2007.

[52] L. Pei, R. Chen, J. Liu, T. Tenhunen, H. Kuusniemi, and Y. Chen. An inquiry-based bluetooth indoor positioning approach for the finnish pavilion at shanghai world expo 2010. In *IEEE/ION Position, Location and Navigation Symposium*, pages 1002–1009, May 2010.

[53] A. N. Pettitt. A two-sample anderson–darling rank statistic. *Biometrika*, 63:161–168, 1976.

[54] Claudio Piciarelli, Gian Luca Foresti, and Lauro Snidaro. Trajectory clustering and its applications for video surveillance. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 40–45. IEEE, 2005.

[55] Lorien Pratt and Sebastian Thrun. Special issue on inductive transfer. *Machine Learning*, 28(1), 1997.

[56] Weijun Qin, Jiadi Zhang, Bo Li, and Limin Sun. Discovering human presence activities with smartphones using nonintrusive wi-fi sniffer sensors: The big data prospective. *International Journal of Distributed Sensor Networks*, 2013:12, 2013.

[57] Xing Xie Yukun Chen Wenyu Liu Wei-Ying Ma Quannan Li, Yu Zheng. Mining user similarity based on location history. In *ACM SIGSPATIAL GIS 2008*. ACM SIGSPATIAL GIS 2008, November 2008.

[58] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 293–304, New York, NY, USA, 2012. ACM.

[59] C. C. Robusto. The cosine-haversine formula. *The American Mathematical Monthly*, 64(1):38–40, 1957.

[60] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[61] Hossein Sarshar and Stan Matwin. Using classification in the preprocessing step on wi-fi data as an enabler of physical analytics. In *International Conference on Machine Learning and Applications*, ICMLA'16. IEEE, 2016. [To Appear].

[62] Suranga Seneviratne, Fangzhou Jiang, Mathieu Cunche, and Aruna Seneviratne. Ssids in the wild: Extracting semantic information from wifi ssids. In *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pages 494–497. IEEE, 2015.

[63] Alexander Strehl, Er Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *In Workshop on Artificial Intelligence for Web Search (AAAI 2000*, pages 58–64. AAAI, 2000.

[64] Venkata N. Padmanabhan Lili Qiu Saikat Guha-Deepanker Aggarwal Venkat Padmanabhan Swati Rallapalli, Krishna Chintalapudi. Physical analytics: A new frontier for (indoor) location research. Technical report, Microsoft Research, October 2013.

[65] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[66] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, Santa Clara, CA, March 2016. USENIX Association.

[67] Venkat Padmanabhan Victor Bahl. Enhancements to the radar user location and tracking system. Technical report, February 2000.

[68] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[69] Glenn Wilkinso. Snoopy: A distributed tracking and data interception framework. [Online; accessed November 28, 2016].

[70] Glenn Wilkinson. Digital terrestrial tracking: The future of surveillance, 2014.

[71] Chenshu Wu, Zheng Yang, and Yunhao Liu. Smartphones based crowdsourcing for indoor localization. *IEEE Transactions on Mobile Computing*, 14(2):444–457, 2015.

[72] Liu Yang, Steve Hanneke, and Jaime Carbonell. A theory of transfer learning with applications to active learning. *Mach. Learn.*, 90(2):161–189, February 2013.

[73] Zheng Yang, Zimu Zhou, and Yunhao Liu. From rssi to csi: Indoor localization via channel response. *ACM Comput. Surv.*, 46(2):25:1–25:32, December 2013.

[74] Moustafa Youssef and Ashok Agrawala. Handling samples correlation in the horus system. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1023–1031. IEEE, 2004.

[75] Shi Zhong. Efficient online spherical k-means clustering. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 5, pages 3180–3185 vol. 5, July 2005.

[76] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.

Appendix A

Documents

A.1 Google Places API Result

Below is a sample result of Google Places API engine for searching *McDonalds* with location constraint 44.645663,-63.577718 (Halifax Downtown) and radius set to 1000 meters.


```
{
   "html_attributions" : [],
   "results" : [
      {
         "formatted_address" : "5201 Duke St, Halifax, NS B3J 1N9, Canada",
         "geometry" : {
            "location" : {
               "lat" : 44.65047929999999,
               "lng" : -63.5763439
            }
         },
}
```

```
{
         "icon" :
            "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
         "id" : "42fe585387148151e5bb8691b749f7989725bf5f",
         "name" : "McDonald's",
         "opening_hours" : {
            "open_now" : true,
            "weekday_text" : []
         },
         "place_id" : "ChIJXW_hEtIjWksR2cKpH4rvFss",
         "price_level" : 1,
         "reference" :
            "CmRSAAAAOOOLgxUPGN2K7HEdkeqYCVY-pD8AOMdbs0GkQNFDSmoulOok1lY...",
         "types" : [ "restaurant", "food", "point_of_interest", "establishment" ]
      },
      {
         "formatted_address" : "5675 Spring Garden Rd G08, Halifax, NS B3J 1G9,
            Canada",
         "geometry" : {
            "location" : {
               "lat" : 44.6426336,
               "lng" : -63.5790724
```

```
            },
            "viewport" : {
               "northeast" : {
                  "lat" : 44.64274159999999,
                  "lng" : -63.5789151
               },
               "southwest" : {
                  "lat" : 44.64230959999999,
                  "lng" : -63.57954429999999
               }
            }
         },
         "icon" :
            "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
         "id" : "83da942fd45851d1c2e97d2ddbf696ffb54a839c",
         "name" : "McDonald's",
         "opening_hours" : {
            "open_now" : true,
            "weekday_text" : []
         },
         "photos" : [
            {
```

```
            "height" : 1989,
            "html_attributions" : [
               "\u003ca
                  href=\"https://maps.google.com/maps/contrib/1104595.../photos..."
            ],
            "photo_reference" :
               "CoQBdwAAAG46GstvoeUt8pTZc5_1X3e0W55gTqATGGnaFuWH4gCGH...",
            "width" : 2970
         }
      ],
      "place_id" : "ChIJP0PJ1DMiWksRU0j_8kfDqAY",
      "price_level" : 1,
      "rating" : 3.2,
      "reference" :
         "CmRRAAAAUseBdnjwv8oKTQz9r98HkDgKQSeJdnVkPKVI978dqmOKaDhQFytVt-k3Yz6m...",
      "types" : [
         "caf\'{e}",
         "restaurant",
         "food",
         "store",
         "point_of_interest",
         "establishment"
```

```
60            ]
61          }
62       },
63       // .... Other results are removed
64    ]
65 }
```