Winter 12-1-2020

A New Set of Wi-Fi Dynamic Line-Based Localization Algorithms for Indoor Environments

Nelson Shaw
shawne@sheridancollege.ca

A New Set of Wi-Fi Dynamic Line-Based Localization Algorithms for Indoor Environments

A Thesis

presented to

School of Applied Computing, Faculty of Applied Science and Technology

of

Sheridan College, Institute of Technology and Advanced Learning

by

Nelson Shaw

in partial fulfilment of the requirements

for the degree of

Honours Bachelor of Computer Science (Mobile Computing)

December 2020

A New Set of Wi-Fi Dynamic Line-Based Localization Algorithms for Indoor Environments

by

Nelson Shaw

Submitted to the School of Applied Computing, Faculty of Applied Science and Technology
on December 13, 2020, in partial fulfillment of the
requirements for the degree of
Honours Bachelor of Computer Science (Mobile Computing)

Abstract

Localization is of great importance for several fields such as healthcare and security. To achieve localization, GPS technologies are common for outdoor localization but are insufficient for indoor localization. This is because the accuracy and precision of the users' indoor locations are influenced by many factors (e.g., multipath signal propagations). As a result, the methodologies and technologies for indoor localization services need to remain continuously under development. A related challenge is the time complexity of the methodologies which impacts the performance of the mobile phones' limited resources. To address these challenges, a new set of fingerprinting algorithms called Fingerprinting Line-Based Nearest Neighbor (FLBNN) is proposed. Furthermore, the new set is compared to other existing Nearest Neighbor-based algorithms. When the deployment of four access points is considered, the FLBNN algorithms outperform several algorithms in terms of accuracy such as Nearest Neighbor version 2, Nearest Neighbor version 4, and Soft-Range-Limited KNN by approximately 17.1%, 7.8%, and 24.1%; respectively. With regards to precision, the new set of algorithms outperforms Path-Loss-Based Fingerprint Localization (PFL) and Dual-Scanned Fingerprint Localization (DFL) by approximately 7.0% and 60.9%; respectively. Moreover, the FLBNN algorithms have a time complexity of O(t * p) where the term t is the number of deployed centroids and the term p is the number of Path Loss exponents. In addition, the new set of algorithms achieves faster run time compared to those for PFL and DFL. As a result, this Thesis improves the cost and reliability of the indoor location services.

Keywords: Indoor Location Services (ILSs), Fingerprinting, Wi-Fi, Path Loss exponent (PLe), and K-Nearest Neighbor (KNN).

Thesis Supervisor: Dr. Tarek El Salti
Title: Professor, School of Applied Computing

Acknowledgments

The authors would like to thank Dr. Tarek El Salti for the support in supervising this Thesis and the professors of the Faculty of Applied Science and Technology at Sheridan.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ability to calculate a user's location indoors is gaining increased attention and demand because of the ubiquity of mobile devices and indoor wireless networks [1]. Indoor localization is a growing research area for engineers and researchers since there are several fields including healthcare, emergency/disaster management, surveillance, security, and industry that benefits from this research area [1] [2]. An example of this research area is the application of locating a patient within a hospital or nursing home [1]. Currently, elderly patients with Alzheimer's or Dementia may have difficulty navigating the indoor environment and may be disoriented and lost. Determining the patient's location by a caregiver is considered vital in order to monitor the individual and to prevent any injuries from occurring [3]. This situation is expected to be an increasing issue within society as patients with Alzheimer's consist of 10% of the population over the age of 65 [4]. An estimation shows that by 2050, elderly people will consist of 21% of the world population [5].

Traditionally, technologies such as GPS are used to determine the location of a user outdoors. However, due to the complexity of indoor environments (e.g., walls and ceilings), these technologies are not sufficient to localize a user indoors [6]. Historically, researchers propose the use of indoor localization technologies such as a security camera system in order to track people [3]. However, the use of a camera may be prohibited due to privacy concerns [3]. A second solution for people within a hospital or nursing home advocates for the use of technology such as RFID tags and wireless receivers in order to track the person's location [3]. For people within their own homes, caregivers may help to monitor their conditions, but camera monitoring is required on a continuous basis which causes a loss in productivity and is not a cost-effective solution for those with the responsibility of monitoring the person [5]. An alternative strategy for monitoring people with a potential illness is the implementation of a home monitoring system using infra-red motion sensors, however, these systems are often intrusive and expensive [5].

New and promising indoor localization systems are based on a Wireless Local Area Network (WLAN) or a Wireless Sensor Network (WSN) [7]. A WLAN is a LAN which utilizes wireless technologies. Whereas, a WSN is a network that consists of sensors that detect wireless signals emanating from a device [7]. In most cases, a Wi-Fi access point or Bluetooth beacon are often used for location-based services indoors. Using access points or beacons, signals from various sources are collected by a target device at a particular indoor location. Based on the signal characteristics, the location of the target device is estimated. Therefore, people may be monitored with a mobile device such as a smart watch and Wi-Fi signals to determine their location indoors, technology which has been implemented [5].

1.1 Problem Background and Motivation

With regards to locating users indoors, there are many challenges to developing an indoor localization system. These challenges include infrastructure cost, reliability, and the ability for the system to adapt to many different environments [8]. The construction of the system is based on various technologies such as Wi-Fi, Bluetooth, Ultra-Wide Band, RFID, and Zigbee in order to estimate the location of a device indoors [9] [10]. The motivation for using Wi-Fi over other technologies is that it is widely deployed for indoor localization due to its low infrastructure cost and high reliability indoors [10] [11]. Upon receiving the signal, different techniques based on the signal's characteristics are used to estimate the user's location. These techniques include the Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and the Receive Signal Strength Indicator (RSSI) [2] [12].

ToA represents the time it takes for a signal to be transmitted from a signal transmitter such as a Wi-Fi access point or a Bluetooth beacon to the user's device [12]. TDoA is a measurement which involves finding the difference in signal arrival time between a pair of access points [2] [13]. Finally, AoA is a measurement that represents the angle of a signal's arrival at an access point transmitted from the target device [12]. However, the problem with using ToA, AoA, and TDoA is that they require additional hardware (i.e., ToA and TDoA require at least three access points and AoA requires two access points) [2] [12]. In addition, the user's device requires a high frequency Analog to Digital Converter (ADC) in order to process the ToA and TDoA signal characteristics which is considered costly [14] [15]. The AoA signals are also highly susceptible to angle multipath, which is common for indoor environments [12].

RSSI represents the strength of a signal that has been transmitted from an access point to the target device. RSSI is a measurement used more frequently than ToA, TDoA, and AoA primarily because of its lower hardware cost and lower power consumption on mobile devices [10]. However, RSSI is still susceptible to multipath fading and noise [7]. As a result, when processing raw RSSI measurements, there is a possibility for "noisy" signals through interference caused by the complex indoor environment [10]. In order to effectively manage "noisy" RSSI measurements, a Kalman filter is used to smooth the RSSI measurements [16].

The indoor localization techniques that use RSSIs include ranging and fingerprinting [9]. Ranging is a technique which involves trilateration [9]. Trilateration involves using three or more access points, which send signals to the target device [10]. However, trilateration is not suitable for indoor localization due to the high complexity of indoor environments [17]. In order for trilateration to be effective, the indoor space must be less spatially diverse and wireless signals should be filtered [9].

Another approach, known as position fingerprinting, is used to improve the indoor localization of a device and functions more effectively within indoor environments that are more spatially diverse [9]. Compared to non-RSSI techniques (e.g., ToA, TDoA, AoA), fingerprinting is not as susceptible to angle multipath and achieves high accuracy without an ADC and clock synchronization among the target device and the access points [1] [2] [11] [12] [13] [14] [15] [18]. The fingerprinting technique consists of two stages known as the offline and online stages [17]. The offline stage involves the collection of RSSI measurements at a grid point in the indoor environment to construct a radio map [10] [17]. This process is repeated for every grid point in the offline stage. The online stage involves the collection of RSSI measurements in real-time and comparing them to those collected in the offline stage at each grid point [19].

One of the first approaches to take advantage of fingerprinting is the RADAR system, which utilizes the Nearest Neighbor algorithm (NN), the K-Nearest Neighbor algorithm (KNN), and Wi-Fi signals in order to estimate the indoor location of a user [20] [21]. However, the average accuracy of the system is approximately 3 m, which is not ideal for accurately estimating indoor locations [20] [21] [22].

Another version of the KNN algorithm referred to as Weighted KNN (WKNN) is based on Statistical Learning Theory [21] [22]. The algorithm calculates the weighted average of RSSI-based distances between a set of selected grid points and the current location [21] [22]. This approach improves the accuracy from 3.12 m to 3.06 m, however, the accuracy is still relatively low [22]. In addition, the accuracy of the approach has not been statistically verified [22]. To improve the accuracy metric, a new weighted fusion fingerprinting algorithm is proposed [23]. With the proposed algorithm, the accuracy has been improved to approximately 1.5 m [23]. However, the precision metric was not considered in the performance evaluation of the algorithm.

Another work by Zhang et al. [24] implements the Path-Loss-Based Fingerprint Localization algorithm (PFL) and Dual-Scanned Fingerprint Localization algorithm (DFL) in order to improve the accuracy and precision of indoor localization [24]. However, the time complexities of these algorithms were not considered [17] [24].

A more recent work discusses the use of a Soft-Range-Limited KNN algorithm (SRL-KNN) in order to address both the time complexity and accuracy [21]. The conducted experiment demonstrates that the SRL-KNN algorithm achieves an accuracy of 0.66 m on average, which is improved compared to that for KNN with the same time complexity [21]. While the work demonstrates that the algorithm achieves a high accuracy, the precision of the algorithm has not been evaluated. Therefore, all these challenges define the following problems:

1. How to achieve high accuracy?
2. How to achieve high precision?
3. How to lower time complexity?

1.2 Research Question

Does the line-based shifting of the grid points improve the indoor localization metrics?

With regards to indoor localization, there are three key metrics to consider. First, the accuracy metric refers to the distance error between the estimated location and the actual location of a target device. By improving the accuracy, the algorithm more effectively localizes the location of the target device in indoor environments. Accuracy is important to consider as it represents how close the predicted location measurement is to the actual location of the target. Accuracy is calculated using the Euclidean Distance, which represents the distance between two points in a grid space.

Another metric is precision, which refers to the distance errors (i.e., Euclidean distances) calculated between the localized and actual positions. Improving the precision means that the calculated positions are more concentrated around a particular area rather than more greatly distributed in an indoor space. Precision is determined by the standard deviation of all the Euclidean Distance values between each calculated location to a particular test location in an indoor space.

Time complexity refers to the run time complexity of an algorithm. The Big O Notation, which is a common metric that computes time complexity, must be considered for fingerprinting algorithms because of the limited resources of mobile devices such as the lifetime battery.

The method of indoor localization using the line-based shifting method is captured in the Fingerprinting Line-Based Nearest Neighbor algorithm (FLBNN). Similar to other fingerprinting algorithms, the FLBNN algorithm requires a radio map constructed in the offline phase. This stage of fingerprinting involves collecting RSSI measurements at each grid point in order to create a radio map. During the online phase of fingerprinting, the new algorithm calculates the distance between each grid point measured on the radio map and the target location. The algorithm then selects the Point S (i.e., grid point or centroid) based on the least distance towards the location of the target. The Point S has four surrounding grid points where two of the grid points, G1 and G2, are selected since they are the closest to the target. A Midpoint M is then calculated between these two closest grid points (e.g., Grid Point G1 and Grid Point G2). The algorithm shifts the closest Grid Point G1 or G2, Midpoint M, or Point S closer to each other. As a result, the Centroid Q is calculated between these three points (Point S, Midpoint M, and the closest Grid Point G1 or G2) and is considered the localized position for the target device.

Given the research question, we propose three possible answers. For the 'Yes' answer, a possible explanation is mentioned as follows. The accuracy improves due to the ability of the algorithm to shift the calculated location possibly closer to the target location. In comparison, the precision improves because the distance errors from the target location to the calculated locations are found to be within the same region of space in the indoor environment. Therefore, there are low variations in distance errors. Finally, the time complexity improves because the algorithm selects a subset of grid points from the super set of grid points than scanning all of the grid points.

For the 'No' answer, a possible explanation is mentioned as follows. The accuracy does not improve because the estimated location is possibly shifted away from the target location by the algorithm. The precision does not improve because the fluctuation of RSSI values cause the distance errors to be calculated within different regions of the indoor environment. Finally, time complexity increases because the algorithm checks all the grid points instead of a subset of them.

For the ‘Yes, Sometimes’ or ‘No, Sometimes’ answer, a possible explanation is as follows. The complexity of the indoor environment (e.g., signal obstruction, multipath fading) results in fluctuations of RSSI values [7]. As a consequence, the variation in RSSI values impacts the calculation of the localized positions and thus impacts accuracy and precision.

1.3 Contribution Overview

The methodology proposed for the Thesis is a quantitative research approach which involves modifying K-Nearest Neighbor (KNN) and comparing the algorithm with other fingerprinting-based algorithms. The contributions of this work are listed as follows.

1. The Fingerprinting Line-Based Nearest Neighbor (FLBNN) algorithm improves accuracy compared to Nearest Neighbor version 4 (NNv4), Nearest Neighbor version 2 (NNv2), Dual-Scanned Fingerprint Localization (DFL), and Soft-Range-Limited KNN (SRL-KNN) when all four Access Points (APs) are used. FLBNN also improves accuracy compared to NNv2, NNv4, Path-Loss-Based Fingerprint Localization (PFL), and the DFL algorithms when a subset of APs are used. Accuracy is a critical metric because it represents the proximity of the location measurement to the actual location. Achieving high accuracy is crucial for several applications such as emergency applications for finding a user in a specific room [25].

2. The FLBNN algorithm improves precision compared to PFL and DFL when all four APs or a subset of APs are used. Additionally, FLBNN also achieves similar precision to NNv4 and SRL-KNN when all APs are used or a subset of APs are used. Precision must be considered because it represents a continuous evaluation of any localization algorithm [10]. In other words, the algorithm will be able to reliably calculate the estimated locations within a region over a period of time.

3. The FLBNN algorithm improves time complexity compared to several existing algorithms including PFL and DFL. Complexity is important because low time complexity is necessary to reduce energy consumption for mobile devices [26].

1.4 Thesis Organization

This Thesis is organized as follows. For Chapter 2, we review existing localization approaches along with some preliminaries. In Chapter 3, a new localization algorithm is proposed. Chapter 4 lists the findings of the research and analyzes the data collected regarding the FLBNN algorithm. Finally, we summarize the Thesis, draw out some conclusions, and provide some future research avenues.

Chapter 2

Literature Review

In this chapter, we discuss existing technologies and techniques related to indoor localization. First, we introduce some preliminaries as a background for the existing work and the proposed work. Additionally, we mention some existing techniques and technologies for indoor localization.

2.1 Preliminaries

2.1.1 Technologies

Wi-Fi, Bluetooth, UWB, RFID, Li-Fi, and Zigbee are some types of technologies that can be used to estimate indoor locations [9] [10]. Among these technologies, Wi-Fi is a common technology for indoor localization due to its low infrastructure cost and high reliability indoors [10] [11]. Another key motivation for using Wi-Fi is that the technology has been already implemented in existing infrastructures [10] [11].

2.1.2 Methodologies

There are two types of features that exist for indoor localization that include Receive Signal Strength Indicator (RSSI) measurements and Path Loss exponent (PLe). These RSSI measurements are used to find the strength of a signal from an Access Point (AP). Furthermore, those indicators are commonly measured in decibelmilliwatts (dBm). Often, the farther from the AP the target is, the lower the RSSI value is [27]. RSSI-based algorithms are used more frequently than ToA, TDoA, and AoA because of their low hardware cost and lower power consumption on mobile devices [10].

Compared to RSSI, a path-loss model can also be used to estimate the distance values between each AP and the target node for estimating the target location [9] [28]. A model commonly used to calculate this distance is the Log Distance Path Loss model (LDPL) [9] [28] [29] [30]. The model is described by the following equation:

$$\mathbf{PL(d)} = PL(d_0) - 10nlog(\frac{d}{d_0}) + X_0 \tag{2.1}$$

From the above formula, the term $PL(d)$ represents the Path Loss or alternatively, the RSSI calculated at an arbitrary distance d [9] [28]. The term $PL(d_0)$ is the average Path Loss or RSSI value previously calculated at a known distance d_0 (d_0 is usually one meter) [9] [30]. The term n represents the Path Loss exponent [9] [28]. The Path Loss exponent value often ranges from 2 to 4 for indoor environments but can be selected empirically [29] [31]. Finally, the term X_0 is a gaussian random variable representing shadow fading [9] [28] [32].

2.1.3 Fingerprinting Stages

Using the measurements such as RSSI or path loss, one method of calculating the estimated location is fingerprinting. The fingerprinting technique consists of offline and online stages. The offline phase involves the collection of signal information at a known grid point in the indoor environment. RSSI measurements are collected at each grid point to construct a radio map (see figure 2-1). Each grid point has a set of RSSI values $RSSI_j = (RSSI_1, RSSI_2, \ldots\ RSSI_n)$ where j is the AP index and n is the number of RSSI measurements collected at each grid point during the offline phase.

The online stage involves the calculation of RSSI measurements in real-time and comparing them to the offline stage RSSI measurements at each grid point [19]. Al-

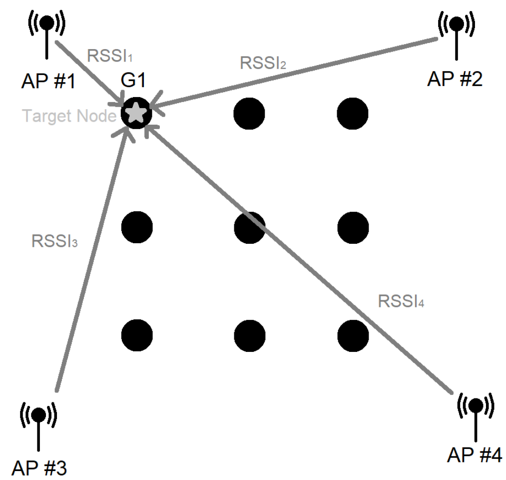

Figure 2-1: Offline Stage Grid Point Map.

though we focus on RSSI-based algorithms in this work, we explore several other techniques for indoor localization including non-RSSI-based techniques in order to cover the breadth of the localization area.

2.1.4 Metrics

Once fingerprinting is used to calculate the estimated location, the metrics used to evaluate the indoor localization capabilities of these algorithms are accuracy, precision, and time complexity. For the first metric, accuracy is the distance error between the location estimated and the actual location. Accuracy is computed using the Euclidean Distance formula in 2D:

$$\mathbf{E}(\mathbf{error}) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{2.2}$$

From the above formula, the pair (x_i, y_i) is the estimated position, and (x, y) is the actual target location. Another metric to consider is precision, which refers to the distribution of the estimated locations and the relative distances between them. Precision is calculated by taking the standard deviation of the accuracy. The formula for precision is as follows:

$$\mathbf{precision} = \sqrt{\frac{\sum_{i=1}^{n}(a_i - a)^2}{n - 1}} \tag{2.3}$$

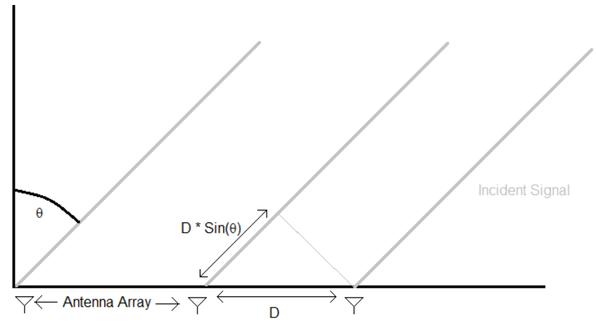

Figure 2-2: Angle of Arrival Technique.

From the above formula, the term a_i is the accuracy of a particular point estimation, the term n is the number of test values collected in the online stage, and the term a is the average accuracy.

2.2 Non-Receive Signal Strength Indicator Techniques

2.2.1 Angle of Arrival

The first non-RSSI technique is Angle of Arrival (AoA). This technique involves the computation of the angle that a signal arrives at the receiver from a target device using antenna arrays [2] [18]. Two APs at a minimum are required to use this technique [2]. Using two incident signal waveforms that are approaching an AP, AoA can calculate the Angle θ of the signal's arrival (see figure 2-2) [1] [2]. When the angle is calculated, a line originating from the AP to an intersection point between the lines of the other APs and in the calculated angle of the signal's arrival can be drawn [2]. Once the lines are created, the user's location can be estimated by the intersection of the lines from every AP [2] [18]. The algorithms used to calculate AoA include Conventional Beamformer and Multiple Signal Classification (MUSIC) [33] [34]. An array of antennas for one AP are separated at an even distance from each other. The formula shown can then be used to calculate the Angle θ [1].

A recent research work regarding AoA is performed by Zheng et al. [34]. The work involves using a weighted AoA approach that assigns a weight to each AoA measurement based on the accuracy of the angle estimation. The motivation is the investigation of varying accuracies of angle calculations in the traditional AoA methods and the application of a weight to the angle using asymptotic variance. The weighted AoA method involves assigning a larger weight to the angle calculation which has the lower asymptotic variance. The methodology for the research involves comparing the algorithm with existing AoA approaches such as the traditional unweighted Least Squares Method (LSM). The main contributions of this work involve the use of AoA, ToA, and asymptotic variance to develop a weighted version of traditional LSM in order to effectively use the accuracy of the angle measured when calculating a user's location indoors. The results of the simulation experiment with two antennae per receiver demonstrate that the median localization error of the weighted AoA is 0.4 m compared to the unweighted LSM with 0.5 m. The weighted AoA method reduces the estimation error by 20% compared to the unweighted method [34]. However, a major flaw with AoA is the additional hardware requirements [1] [12]. For AoA to achieve good accuracy, it must involve Line of Sight (LoS) conditions as it suffers from angle multipath [12] [18].

2.2.2 Time of Arrival

Time of Arrival (ToA) is a technique for indoor localization that involves measuring the travel time of a signal from a transmitter to a receiver [1]. Compared to AoA, ToA does not require LoS conditions for high accuracy [18] [35]. In order to achieve high accuracy for ToA, a minimum of three APs are used to estimate the location [12]. The distance between the transmitter and receiver is calculated using the speed of light and the propagation delay of the signal in NLoS conditions [35] [36] [37]. Figure 2-3 demonstrates the ToA technique with $T1$, $T2$, and $T3$ representing the time each of the three signals spent travelling to each AP.

A work done by Lei et al. [38] proposes a single anchor node system which measures ToA at a continuous rate. The motivation of the study is the use of distance

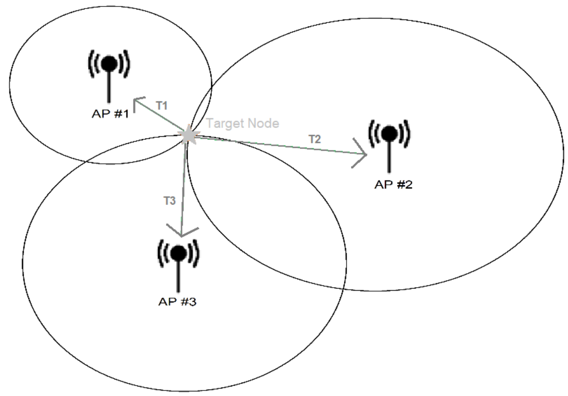

Figure 2-3: Time of Arrival Technique.

lines from a target node to an anchor and the angle between two adjacent lines when the target or anchor moves. Based on these measurements, the researchers were able to devise a system that involves a single AP on a rotating blade to calculate ToA over time while changing positions. As the AP is rotating, the distances between the AP and the target are calculated and a graph representing those distances over time is created as part of an offline stage. During the online stage, the target location is calculated by identifying the change in time intervals for ToA measurements during a rotation period. The change in angle of the target is then calculated using the change in time intervals. In addition, the Support Vector Machine (SVM) classifier is also used to reconstruct the angle between two adjacent distance measurements when there is variation in the LoS conditions. The first experiment involves a LoS scenario while the second experiment involves NLoS. The contributions provided by the paper is that a single AP's ToA calculation method based on UWB is proposed to improve accuracy beyond traditional ToA techniques. Another contribution is the use of SVM to reconstruct the angle estimation in more variable LoS conditions. In the first experiment, the accuracy of the system is less than 0.3 m while the accuracy of the system in the second experiment is less than 0.6 m [12] [38]. However, a flaw with this method is that it requires additional resources such as a motor with a rotating blade in order to accomplish high accuracy. Additionally, another issue is the lack of requirements of at least three APs. Other than the flaws related to the study, the

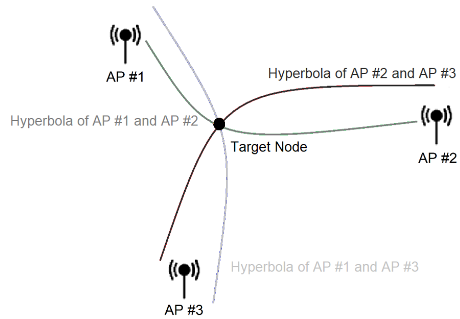

Figure 2-4: Time Difference of Arrival Technique.

main issue with ToA is that the transmitter and the receiver nodes are required to have precise clock synchronization [1] [11] [12] [18].

2.2.3 Time Difference of Arrival

Time Difference of Arrival (TDoA) is a measurement which involves finding the difference in signal arrival time between two APs [2] [13]. Unlike ToA, TDoA only requires synchronization between the receivers [1] [11] [12]. Using the difference in signal time arrival, the TDoA measurement defines a hyperbola between a pair of APs (see figure 2-4) [2]. Three hyperbolas are created, and the location of a user can be estimated by finding the intersection between them [1] [2].

A recent work proposed by Xie et al. [11] shows a coarse estimation method for indoor localization using TDoA in conjunction with RSSI to mitigate the effects of multipath with Wi-Fi signals. The motivation is to examine the effects of multipath and to create an indoor localization method that reduces its effects. The methodology is focused on modification of the traditional TDoA technique using RSSI in addition to testing the new method with a prototype application. The modified algorithm first estimates the mobile node position using cross-correlation and quadratic filling to determine the distance between an AP and the target for two APs. The algorithm then uses RSSI to estimate the polarity of the proposed Multipath Signal Interference (MPSI) metric used to measure the impact of multipath for each AP. MPSI is used to compensate for the signals received by the receivers. Finally, the TDoA calculation is performed using cross-correlation and quadratic filling again based on the compensated signals. The major contribution is that a new TDoA method is proposed to effectively manage multipath interference compared to traditional TDoA methods. In the experiment, the maximum position error of the proposed method is 0.3 m in the 90th percentile compared to the traditional TDoA method which has 1.5 m accuracy in the 90th percentile [11]. Similar to the flaws with the ToA method, TDoA requires clock synchronization among receivers in order to calculate the difference in time between the received signals [11] [18]. Additionally, the user's device requires a high frequency Analog to Digital Converter (ADC) in order to process the ToA and TDoA signal data, which could be costly for a mobile device [14] [15].

2.3 Receive Signal Strength Indicator Techniques

This subsection focuses on the trilateration and fingerprinting techniques. The fingerprinting technique includes algorithms such as Nearest Neighbor (NN), K-Nearest Neighbor (KNN), Weighted K-Nearest Neighbor (WKNN), a weighted fusion-based algorithm, Path-Loss-Based Fingerprint Localization (PFL), Dual-Scanned Fingerprint Localization (DFL), Nearest Neighbor version 2 (NNv2), Nearest Neighbor version 3 (NNv3), Nearest Neighbor version 4 (NNv4), and the Soft-Range-Limited KNN algorithm (SRL-KNN) [17] [21] [22] [23].

2.3.1 Trilateration

Trilateration is a technique which involves the use of three or more APs to send signals to a mobile device [10]. Trilateration is based on RSSI, and therefore, does not require clock synchronization or a high frequency ADC [10] [11] [14]. After the signals have been received by the mobile device, the signal information may be used to calculate the spatial distance between an AP and the target device (see figure 2-5). A circle can be formed around each AP. The radius of the circle is the calculated distance between the AP and the target. The intersection of the three circles are calculated to be the estimated position [10]. Each distance value and the known coordinates of each AP are used in the Euclidean Distance formula which results in the calculation

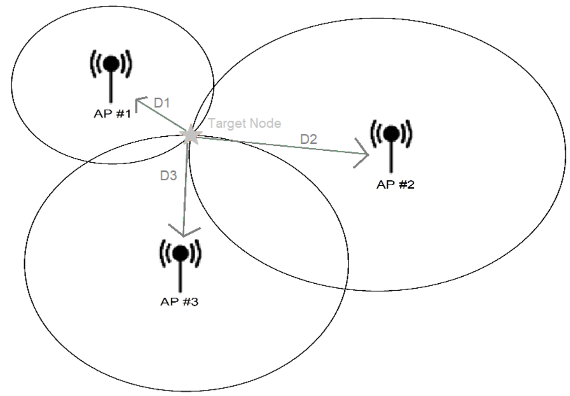

Figure 2-5: Trilateration.

of the target location coordinates [28]. Trilateration uses the Log Distance Path Loss model (LDPL) to calculate distance between each AP and the target node from RSSI values [9] [28] [29] [30]. Figure 2-5 provides an illustration of trilateration with $D1$, $D2$, and $D3$ representing the distances between the target node and each AP.

A recent work from Tong et al. [27] involves the use of a modified trilateration algorithm to estimate locations. The motivation is to improve the relationship between distance and RSSI by using LDPL and curve fitting in order to measure a distance over time graph used by a modified version of trilateration. First, LDPL is used to calculate the distance using RSSI. Afterwards, the distances are calculated as a curve over time and are improved using curve fitting over a set of time slots on the curve. The modified trilateration algorithm, known as Distance Ratio Location (DRL), is based on the following information: four beacon nodes, the distances between them, and their coordinates. The algorithm uses this information in order to determine a vector based on a pair of beacons and the estimated target location in order to create a continuous location curve. Finally, the Kalman filter optimizes the raw RSSIs. The contributions of the work include curve fitting in order to correct the distance between a target node and a beacon node and a modified version of trilateration to better calculate indoor locations. DRL achieves an accuracy of 3.5 m indoors compared to the Maximum Likelihood Estimation algorithm with an accuracy of 17 m [27]. One of the flaws in the research is the low accuracy achieved by the system. Another drawback in the algorithm is that it has increased time complexity for indoor localization. However, a major problem with trilateration is that the indoor space must be less spatially diverse, and the wireless signals used for trilateration are required to be robust [9]. Additionally, LoS conditions are necessary for trilateration to achieve high accuracy [39]. When the distance between the target and an AP according to RSSI is estimated, this distance is considered challenging to calculate when the signal is affected by various obstructions (e.g., walls and ceilings) within the indoor environment.

2.3.2 Fingerprinting-Based Algorithms

Fingerprinting is a method which is more effective for environments that are more spatially diverse [9]. Compared to trilateration, it does not require LoS conditions for high accuracy [39]. Fingerprinting collects the RSSI data at grid points within an indoor space. Fingerprinting often involves Nearest Neighbor (NN) or K-Nearest Neighbor (KNN). NN consists of an offline stage similar to fingerprinting [20]. The online stage of NN selects the point from the list of grid points that has the minimum distance to the location of the target using the Euclidean distance [20]. However, the performance of NN depends on the number of points from the list of grid points [17]. KNN calculates an estimated centroid point between K selected grid points [20] [40] [41]. However, both algorithms can lead to inaccuracies due to multipath interference associated with RSSI and can impact the grid point selection process [17].

a. RADAR System

The NN and KNN algorithms are adopted by the RADAR system, one of the first systems to take advantage of fingerprinting. The motivation of the system is that Bahl and Padmanabhan [20] explored the use of radio frequency networks over infrared to address range, scalability, and deployment. The main contribution is a system that can take radio frequency signals and estimate a location using NN or KNN. Their approach uses the NN algorithm. Their first experiment involves the use of an empirical method in which one test location is selected at random. The empirical method shows a 2.94 m error rate at the 50th percentile. They also applied the KNN algorithm to the empirical approach in another experiment. KNN (with 2-4 nearest neighbors averaged) has an accuracy of 2.13 m in the 50th percentile [20]. The accuracy of the system is not substantial for indoor localization.

b. Weighted K-Nearest Neighbor

To improve the accuracy of RADAR, another study focuses on the use of Statistical Learning Theory (SLT) applied to indoor localization to determine functional dependencies between signal strength and position. The focus of the study is the discovery of a strong statistical dependence between signal strength and position. The study is also motivated by certain factors such as the passive measurement of RSSI, the lack of additional equipment other than APs and a mobile device, and ensuring the offline stage is conducted as fast as possible. The fingerprinting algorithm used in the paper is Weighted KNN (WKNN) [22], which involves selecting the K-nearest neighbors and calculating the estimated location by taking the average of the K neighbor's locations. The average is then weighted with the inverse of the distance between two signal strength values (the signal strength value from the offline stage of a neighbor and the online stage RSSI value). The time complexity of WKNN is O(r *(a + d)) offline and O(r * a * log(k) + d) online where the term r refers to the number of observations, the term a is the number of APs, the term d is the number of physical dimensions, and the term k is the number of nearest neighbors [22]. The primary contributions include using a SLT technique to determine a user's location and the use of WKNN to improve upon the KNN algorithm used in RADAR. Compared to KNN with an accuracy of 3.12 m, the WKNN algorithm has an accuracy of 3.06 m [22]. The primary flaw in the work is that the accuracy and time complexity of WKNN is not statically significant compared to KNN. Precision was also not mentioned within the work.

A more recent study by Ninh et al. [42] also evaluates the WKNN algorithm in a controlled environment. The study demonstrates that the WKNN algorithm has an average accuracy of 1.5 m under those conditions [42].

c. Weighted Fusion Algorithm

In order to improve accuracy, Ma et al. [23] proposes a new fingerprinting algorithm based on weighted fusion. The motivation is to improve accuracy by combining an improved Euclidean Distance calculation and a joint probability calculation using weighted fusion. The offline stage of the algorithm involves collecting Wi-Fi signals at given grid points and processing the original signal information depending on the presence of different types of errors. Afterwards, each grid point is fingerprinted with the average value of the original signals, the standard deviation of the original signals, and the average value of the processed signals. In the online phase, the fingerprints are first filtered based on the MAC addresses of the APs with a stronger signal strength than the average signal strength of all APs within the target node area. Then, using the filtered grid points, the improved Euclidean Distance involves calculating a temporary set of coordinates using the averages of the processed signals and standard deviation values of the original signals. After the first coordinates are computed, the second coordinates are calculated using a joint probability algorithm. The algorithm utilizes the average and standard deviation of the original signal strength values for the target and grid points. Finally, a weighted fusion algorithm is used along with the lowest Euclidean distance, the highest joint probability value, and the two coordinates in order to estimate the target location [23].

The major contribution is improving accuracy and time complexity using the weighted fusion algorithm compared to WKNN and joint probability. The proposed algorithm has a 1.54 m accuracy while the WKNN algorithm has a 1.66 m accuracy [23]. However, the improved accuracy of the proposed algorithm was not statistically verified and the precision was not considered in the algorithm.

d. Path-Loss-Based Fingerprint Localization and Dual-Scanned Fingerprint Localization

To consider precision, Zhang et al. [24] introduces the Path-Loss-Based Fingerprint Localization (PFL) algorithm and the Dual-Scanned Fingerprint Localization (DFL) algorithm [24]. The motivation is to improve precision in a resource limited environment compared to existing algorithms using fewer signal sources with PFL. The motivation also includes DFL in order to improve accuracy. PFL takes into account of the transmission and receiver gains, the standard deviance of signal fading, and the distance between a particular grid point and an AP in order to provide a more substantial fingerprint map. During the offline stage, the RSSI values are processed into PLe values, which act as the main feature for the grid points. In this case, the NN algorithm is used to calculate the estimated position during the online stage using PLe values instead of raw RSSIs. DFL calculates a distance threshold and a received signal strength threshold in order to filter the grid points for selecting the nearest neighbor in the online stage. The main contribution of the work is PFL, which improves precision using a path loss factor in a limited signal source environment. Another contribution is DFL, which has a higher accuracy than PFL. The results have demonstrated that PFL and DFL improve the precision of locating a user compared to NN and KNN. When using six APs, PFL has shown to have 68% of the samples with a precision of less than 5 m. DFL has 78% of samples showing precision under 5 m. Based on the cumulative distribution function used to examine accuracy, the PFL algorithm achieves a localization error of around 4 m in the 50th percentile while the accuracy of DFL achieves a localization error of around 3 m in the 50th percentile when using six APs [24]. However, the accuracy of the algorithms needs to be improved, especially PFL since it is susceptible to outliers. Additionally, time complexity is a factor which was not considered in the work.

e. Nearest Neighbor version 2, 3, and 4

To examine time complexity and improve accuracy, El Salti et al. [17] propose the NNv2, NNv3, and NNv4 algorithms [17]. The motivation is to improve accuracy compared to PFL, DFL, NN, and KNN. The NNv2 algorithm chooses the closest centroid of four grid points in the online stage (see figure 2-6). The complexity of NNv2 is O(t * m), where the term t refers to the number of deployed centroids and the term m refers to the number of RSSIs collected in the offline phase. Also, t

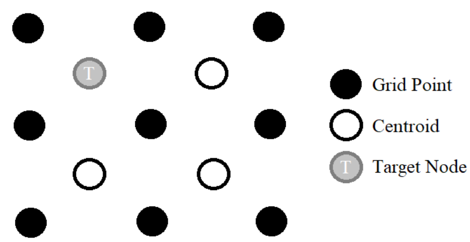

Figure 2-6: Nearest Neighbor Version 2.

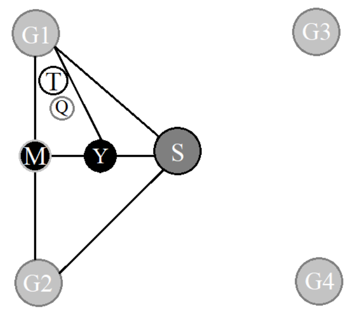

Figure 2-7: Nearest Neighbor Version 3.

= (l - 1) * n where the term l refers to the rows (i.e., number of rows) that hold some grid points and the term n refers to the centroids (i.e., number of centroids) between consecutive rows. The NNv3 algorithm in its online stage chooses the four nearest grid points to the target node (see figure 2-7). If there exists equal distances to the target, the centroid of the grid points is the calculated location. Otherwise, the algorithm chooses the two closest grid points and the Centroid S of the four grid points to calculate the location. If the distances of the two closest grid points are identical, then the algorithm calculates the Centroid Y between the two grid points and S as the estimated location. Otherwise, a midpoint between the two grid points is calculated. The Centroid Q between Centroid Y, the closest grid point, and the midpoint is calculated to be the localized point. NNv4 is similar to NNv3, but the algorithm selects the nearest centroid first with its four nearest grid points instead of four grid points anywhere in the space.

The contributions of the study are three new Bluetooth fingerprinting algorithms that improve both accuracy and precision. The study also compares the algorithms in terms of complexity. The NN, KNN, PFL, NNv2, NNv3, and NNv4 algorithms are compared. NNv4 achieves the highest accuracy with less than 3 m. NNv2 and NNv4 have higher accuracy compared to PFL, NN, and KNN. NNv4 achieves the highest precision of less than 1 m while NNv2, NNv3, and NNv4 achieve better precision than both PFL and NN [17]. However, the accuracy of indoor localization can be improved, as well as the complexity of NNv4.

f. Soft-Range-Limited K-Nearest Neighbor

A more recent work by Hoang et al. [21] proposes a Soft-Range-Limited KNN algorithm (SRL-KNN) [21]. The motivation is to address existing issues with KNN such as the similarities between fingerprints and the variation in RSSI. SRL-KNN utilizes the user's previously calculated position to consider the speed of the user while moving. A circle can be formed around the previous location and the neighbors near that location are more likely to be selected as a K-nearest neighbor. The user's location is then determined through a weighted average of the selected neighbors. The contribution is an algorithm that achieves high accuracy while the target device is stationary and moving. While the target device is stationary, the SRL-KNN algorithm has an accuracy of 0.60 m compared to RADAR, KNN, and WKNN which have accuracies lower than 1.20 m. Another experiment has been performed where the movement of the target device is considered. The authors describe the use of an RSSI histogram algorithm in order to achieve a 0.66 m accuracy with SRL-KNN. The proposed algorithm also has the same time complexity as KNN [21]. A major flaw is that precision was not captured in their work. SRL-KNN is not effective if the previous location calculated is inaccurate. Additionally, a recent study by Zhang et al. [43] demonstrates that even though the SRL-KNN algorithm has high accuracy, the same numerical difference in RSSI fingerprint values between two grid points may not equal the same geometric difference compared to another similar set of two grid points [43]. This is because the radio map for the SRL-KNN algorithm uses the mean RSSI and mean difference in RSSI for fingerprinting features. The experimenters worked with an RSSI histogram to smooth the RSSI values. However, RSSI values for fingerprints can fluctuate, and thus, the fingerprinting values between two grid points may be similar to each other [21].

2.4 Summary

The non-Receive Signal Strength Indicator (RSSI) techniques include Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA). The AoA technique involves the calculation of a signal's arrival angle in order to estimate the user's location but suffers from additional hardware requirements and angle multipath [1] [2] [12] [18]. ToA involves measuring the travel time of a signal from a transmitter to a receiver but requires additional resources and precise time synchronization between the two devices [1] [11] [12] [18]. TDoA involves finding the difference in signal arrival time between two Access Points (APs) but requires a high frequency Analog to Digital Converter (ADC) to process the signal [2] [13] [14] [15].

The RSSI-based techniques include trilateration and fingerprinting [10]. Due to the advantages of RSSI-based algorithms, including its low hardware cost and lower power consumption, this research focuses on these algorithms [10]. Trilateration uses three APs and creates circles around them to find the intersection point but requires additional infrastructure and is not effective in spatially diverse environments [9] [10].

Fingerprinting is an effective technique for indoor localization. Compared to non-RSSI techniques, it is not as susceptible to angle multipath and can achieve higher accuracy without an ADC and clock synchronization [11] [12] [14] [18]. Compared to trilateration, the fingerprinting technique is more suited for the spatially diverse indoor environments [9] [10] [17]. Among the fingerprinting techniques, the NN and KNN algorithms suffer from multipath interference [17]. RADAR and WKNN do not have sufficient accuracy while WKNN has high time complexity [20] [22]. The weighted fusion algorithm does not consider precision [23]. PFL and DFL have relatively low accuracy [24]. NNv2, NNv3, and NNv4 can be improved in terms of their accuracy [17]. Finally, SRL-KNN has potential inaccuracies in locations calculated in previous runs [21].

The fingerprinting algorithms are displayed in Table 2.1 with their accuracy, precision, and time complexity. Time complexity represents the run time complexity of an algorithm and is measured using the Big O Notation. As shown in the table, the time complexity of NN, KNN, and SRL-KNN is O(g * m) where the term g refers to the number of grid points and the term m is the number of RSSI values collected offline [17] [20] [21] [44]. The numbers in the table may change depending on the environments and practices used for Indoor Location Services. Based on these challenges, we propose and efficient algorithm that improves the accuracy and precision for indoor localization in order to effectively estimate the location of a target indoors. The time complexities for PFL and DFL [24] were not considered in their respective research, and therefore, the run time of these two algorithms are evaluated in our study. In addition, the precision of WKNN and SRL-KNN are also investigated in our study as they were not mentioned in their original research [21] [22].

Table 2.1: Fingerprinting-Based Algorithms

Algorithm	Accuracy	Precision	Time Complexity
NN	Around 2.94m - 4m [17] [20]	Around 2m [17]	O(g * m) [20] [44].
KNN	2.13m, 3.12m, and 3.25m [17] [20] [22]	Around 1.3m [17]	O(g * m) [20] [44].
WKNN	1.66m - 3.06m [22] [23]	X	O(r *(a + d)) offline and O(r * a * log(k) + d) online where r is the number of observations, a is the number of APs, d is the number of physical dimensions, and k is the number of nearest neighbors [22].
Weighted Fusion	1.54m [23]	X	X
PFL	Around 3.5m-4m [17] [24]	Under 5m [17] [24]	X
DFL	Around 3m [24]	Under 5m [24]	X
NNv2	Around 3m [17]	Around 1.3m [17]	O(t * m) where t refers to the centroids and m refers to the RSSIs [17].
NNv3	Around 3.75m [17]	Around 1.9m [17]	O(l * n * m + l * m) where l refers to the rows and n refers to the centroids between consecutive rows [17].
NNv4	Around 2.75m [17]	Around 0.9m [17]	O(t * m) where t refers to the centroids and m refers to the RSSIs [17].
SRL-KNN	0.60m [21]	X	O(g * m) [21].

X:Unknown

Chapter 3

Methodology

In this chapter, a new indoor localization algorithm is proposed. Our proposed algorithm and its properties are discussed. Furthermore, a simulation model is designed and used to compare the proposed algorithm against existing Nearest Neighbor-based algorithms [17] [21] [22] [24]. Our proposed algorithm and simulation model are implemented in the next chapter.

3.1 The Fingerprinting Line-Based Nearest Neighbor Algorithm

Essentially, the Fingerprinting Line-Based Nearest Neighbor (FLBNN) algorithm is an extension of the K-Nearest Neighbor (KNN) algorithm. In order to implement the FLBNN algorithm, an offline stage needs to be performed. The offline stage involves the measurement of Receive Signal Strength Indicator (RSSI) measurements at known positions or grid points in the indoor environment. This establishes a radio map. At the online phase, FLBNN is executed. Algorithm 1 lists the detailed steps of the FLBNN algorithm in the online stage. From Lines 1 to 8, the FLBNN algorithm first chooses the nearest Centroid S among all the deployed centroids (see figure 3-1). In figure 3-1, the algorithm chooses the nearest Centroid S when the target point T is localized.

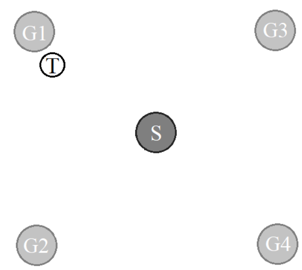

Figure 3-1: FLBNN Step 1.

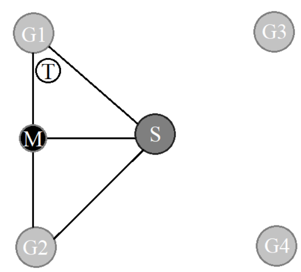

Figure 3-2: FLBNN Step 2.

After the nearest Centroid S is chosen, in Line 9, the algorithm then chooses the two closest grid points where the constructed line between the grid points does not intersect the Centroid S (see figure 3-2). The algorithm calculates the Midpoint M between the two closest grid points and chooses the nearest grid point (e.g., Grid Point G1).

Despite the similarities between FLBNN and Nearest Neighbor version 4 (NNv4), the following steps of FLBNN demonstrate its unique properties. Once the midpoint is calculated, between the nearest Grid Point G1, Centroid S, and Midpoint M, the algorithm has the ability to displace one or two of these points towards either G1, S, or M by a certain percentage. For example, the algorithm can displace G1 towards S or the algorithm can displace M and S towards G1. If Grid Point G1 is displaced by 100% towards Centroid S, Grid Point G1 is directly shifted to the same coordinates as Centroid S. Another example is that Grid Point G1 can be shifted halfway towards Midpoint M when the displacement percentage is equal to 50%.

From Line 10, the algorithm demonstrates the calculation of a set of points N, which has the nearest Grid Point G1, the Centroid S, and the Midpoint M, with one or two of those points displaced by a certain percentage amount (see figure 3-

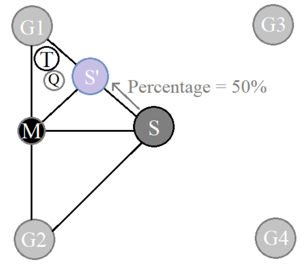

Figure 3-3: FLBNN Step 3.

3). Essentially, the term s represents a set of points that is shifted by a percentage. The set of points s can include S, M, or G1. The set of points specified in s are shifted towards the point r where this point does not equal the point(s) s. The term r represents either G1, S, or M. As a result, the set of points N is formed to have the updated points. Finally, at Line 11, the algorithm will calculate the Centroid Q as the estimated location between the shifted point(s) and the other fixed point(s). In figure 3-3 for example, Centroid S is shifted by 50% towards the nearest grid point, which is represented by point S'.

The time complexity of FLBNN is represented by O(t * p) where the term t refers to the number of deployed centroids and the term p refers to the number of every Path Loss exponents (PLe) calculated from the RSSI measurements during the offline stage.

Algorithm 1: Fingerprinting Line-Based Nearest Neighbor

Input : c - the number of centroids in the grid point space, n - the number of access points, T - the target node at an unknown coordinate, s - a set of one or two points to shift (either Centroid S, Midpoint M, or the closest Grid Point G), r - the point that s will be shifted towards, and dp - the displacement percentage. Each centroid is surrounded by four Grid Points G_1, G_2, G_3, and G_4.

Output: Q - the estimated location coordinates.

```
1  let minDistance = 0;
2  foreach i of c do
3      dist[i] = Distance(T, i);
4      if minDistance > dist[i] then
5          minDistance = dist[i];
6          S = i;
7      end if
8  end foreach
   /* G_i is the first closest grid point and G_j is the second closest.       */
9  M = MidDistance(G_i, G_j);
   /* N represents a set of three points [G_i, S, and M] after the points in s are shifted.  */
10 N = ShiftPoint(s, r, dp);
11 Q = Centroid(N);
12 Return Q;
```

FLBNN has some significant properties. We mention them along with their possible explanations as follows.

1. Hypothesis 3.1: The FLBNN algorithm has higher accuracy compared to those for some existing localization algorithms (i.e., Nearest Neighbor-based algorithms), because the algorithm shifts the closest grid point, the midpoint, or Centroid S possibly closer to the target node. The assumption made here is that the target is not located at a grid point, centroid, or at a midpoint. The algorithm chooses the Centroid S that is the closest to the target and the two closest grid points. In one case, the algorithm shifts the Centroid S towards one of the two grid points where the grid points are closer to the target than the Centroid S. In another case, the algorithm shifts the nearest one of the two grid points closer to Centroid S where S is closer to the target than both grid points. Therefore, the calculated Centroid Q is closer to the target, and thus, the accuracy of the algorithm is improved.

2. Hypothesis 3.2: The FLBNN algorithm has higher precision compared to those for some existing localization algorithms (i.e., Nearest Neighbor-based algorithms) because the algorithm calculates the estimated locations within a particular area of the space (i.e., triangle). The Centroid Q is calculated within a smaller area between the closest grid point, the midpoint, and the Centroid S after displacement. Thus, the distance errors are calculated within that area. As a result, there is less variation in distance errors.

   In other words, one possibility of high precision is that one of the closest grid points is shifted towards Centroid S. In one run of the algorithm, two closest Grid Points G1 and G2 are chosen and the estimated Location Q is calculated. In the next run of the algorithm, two different closest grid points are chosen (e.g., G3 and G4) compared to G1 and G2 which were chosen in the previous run of the algorithm. The estimated updated Location Q is then calculated. In both runs, the algorithm can move one of these grid points (e.g., G1 or G2 in the first run and G3 or G4 in the second run) closer to the Centroid S. Therefore, the estimated Location Q and the updated Location Q are closer to each other since they are both near Centroid S.

Another possibility is that Centroid S is shifted closer to the closest Grid Point G1. In certain cases, the second closest grid point is different in the next run of the algorithm. For example, under the assumption that G1 is the closest grid point in both runs of the algorithm, G2 is the second closest grid point in the first run. The estimated Location Q is calculated in the first run. In the second run of the algorithm, G3 is the second closest point and the estimated updated Location Q is calculated. If that is the case, FLBNN shifts the Centroid S towards the nearest Grid Point G1 each run. Therefore, the estimated Location Q and the updated Location Q are closer to each other because they are close to the nearest Grid Point G1. As a result, the precision of the algorithm is improved.

3.2 Performance Evaluation

3.2.1 Simulation Model and Performance Metrics

The simulation model that is used to verify the important properties of the Fingerprinting Line-Based Nearest Neighbor (FLBNN) algorithm involves a Google Pixel phone with four Wi-Fi Access Points (APs). The model is based on a 2.5 m by 12.5 m area with 12 grid points and 5 centroids in room S144 at the Sheridan Trafalgar campus (see figure 3-4). The distance between each grid point is 2.5 m from each other. The distance between each centroid is also 2.5 m. Each AP is distanced at least 22 feet or 6.71 meters away from each other. The design is based on the Cisco best practices for AP spacing for indoor localization and based on the Sheridan/TELUS project [45].

Existing RSSI datasets are collected on September 7th, 2019 and are used for the offline and online stages. During the offline stage (see figure 3-5), the target device is stationed at each grid point or centroid and collects 35 filtered RSSI measurements

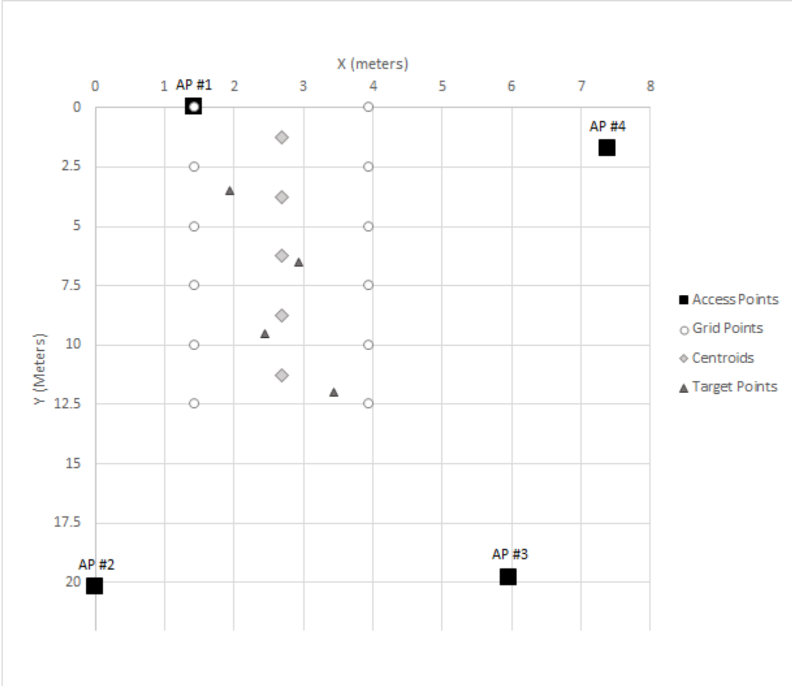

Figure 3-4: Simulation Model Map.

(i.e., Kalman filter). These 35 RSSI measurements for each of the four APs are used to calculate a single PLe value for each AP. Therefore, each grid point (or centroid) are fingerprinted with four PLe values [24].

There are two types of offline stages. The first type is based on the available four APs. Whereas, the second type is based on an AP selection method. For this type, a confusion matrix is used to find the best possible combination of APs that has the highest accuracy. In other words, for the same deployment of grid points and centroids, a number of zones has been created (see figure 3-4). In order to evaluate which AP combination achieves the best accuracy, the grid point space is divided into 2.5 m by 2.5 m zones where each zone contains a centroid in the middle of it. For each AP combination, the target device is stationed in the center of a zone (or each centroid), and 35 RSSI values were measured in each zone to determine which zones the algorithm predicted the target device to be in. The confusion matrix demonstrates the percentage of correctly and incorrectly classified zones. The accuracy value calculated refers to the percentage of correctly classified zones when the target device is standing in each zone. The work performed regarding these matrices is based on results provided from the Sheridan/TELUS project.

In the online stage, the four randomized test locations (1.937, 3.5), (2.437, 9.5), (2.937, 6.5), and (3.437, 12) are used to measure the accuracy and precision for the following algorithms:

a. K-Nearest Neighbor (KNN) [17],

b. Weighted KNN (WKNN) [22],

c. NN version 2 (NNv2) [17],

d. NN version 3 (NNv3) [17],

e. NN version 4 (NNv4) [17],

f. Path-Loss-Based Fingerprint Localization (PFL) [24],

g. Dual-Scanned Fingerprint Localization (DFL) [24] and,

h. Soft-Range-Limited KNN (SRL-KNN) [21].

Additionally, the run time of the FLBNN algorithm is compared to PFL and DFL. The online stage process is shown in figure 3-6. The reason for the comparison is because each algorithm is a Nearest Neighbor-based algorithm. During the online stage, the Midpoint M is shifted by a percentage between 10% to 90% towards Centroid S for the FLBNN algorithm. The FLBNN algorithm is evaluated nine times (one for every 10% displacement up to 90%) for each of the four test locations.

Each of the existing and proposed algorithms are compared in terms of their accuracy, precision, and run time. During the online stage, each algorithm runs 35 times for every test point, and consequently, the metric values for each run are averaged. For the accuracy metric, it refers to the distance error between the location estimated and the actual location. Moreover, precision refers to the distribution of the estimated locations and the relative distances between them.

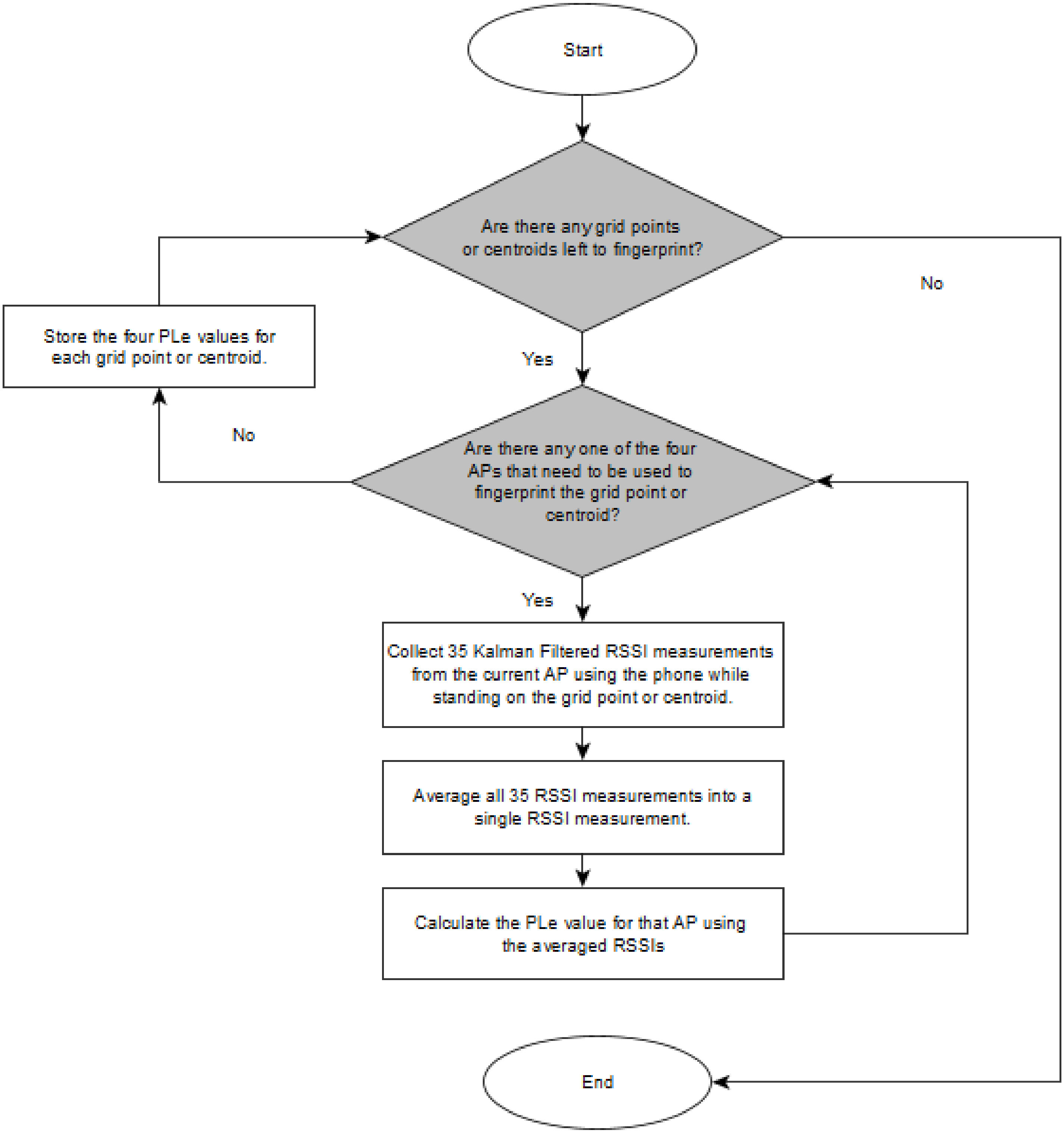

Figure 3-5: Offline Stage Flow Chart.

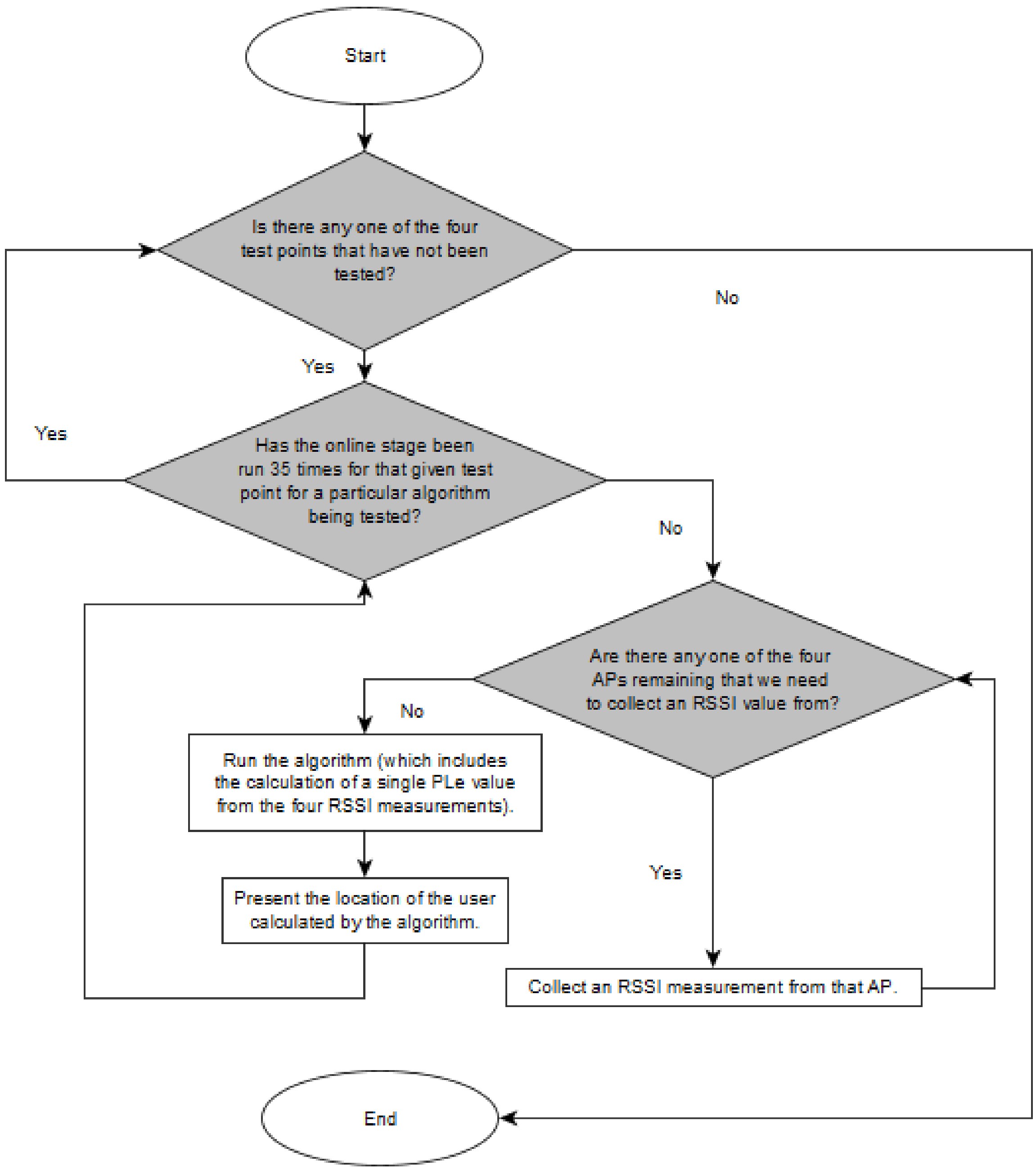

Figure 3-6: Online Stage Flow Chart.

3.2.2 Analysis

The results are analyzed based on either parametric or non-parametric tests depending on the normality testing. In order to perform those tests, the Shapiro-Wilk test is used to calculate the p-value of the normality test for all of the proposed and existing algorithms in terms of accuracy and precision. To determine whether to use parametric or non-parametric tests, the p-value must be greater than α that is equal to 0.05 (i.e., 95% confidence level). If the accuracy or precision values do not follow the normal distribution, a non-parameterized test (e.g., Wilcoxon Signed-Ranks Test) is required to statistically analyze any significant differences between the FLBNN algorithm and the existing algorithms. Otherwise, parameterized tests (e.g., T-Test and ANOVA) are required to analyze the differences in those values. Lastly, for the AP offline selection method, the zones based on subsets of APs are evaluated via the confusion matrices in order to choose the matrix that has the highest accuracy.

Chapter 4

Findings (Analysis and Evaluation)

In this chapter, we explore the results of our experiments and answer our research questions based on our findings. Our results demonstrate the comparisons between the Fingerprinting Line-Based Nearest Neighbor (FLBNN) algorithm and the following algorithms:

1. K-Nearest Neighbor (KNN) [17],
2. Weighted KNN (WKNN) [22],
3. NN version 2 (NNv2) [17],
4. NN version 3 (NNv3) [17],
5. NN version 4 (NNv4) [17],
6. Path-Loss-Based Fingerprint Localization (PFL) [24],
7. Dual-Scanned Fingerprint Localization (DFL) [24] and,
8. Soft-Range-Limited KNN (SRL-KNN) [21].

These comparisons are mainly in terms of accuracy and precision. We further evaluate the localization time for PFL, DFL, and FLBNN. In these results, the FLBNN algorithm shifts the Midpoint M to the nearest Centroid S by a certain percentage between 10% and 90% in these results.

4.1 Results with All Access Points

The existing fingerprinting-based algorithms are first compared to FLBNN in terms of the accuracy metric when all four Access Points (APs) are deployed (see figure 4-1). FLBNN with a displacement percentage of 30% achieves the highest accuracy compared every other fingerprinting algorithm. KNN achieves the highest accuracy among the existing fingerprinting-based algorithms but has a lower accuracy compared to FLBNN. In addition, the DFL algorithm achieves the lowest accuracy. Both FLBNN and KNN are statistically compared in terms of their accuracy by using the Mann-Whitney U test for non-parametric data (see Table 4.1). From this table, we observe whether the difference in accuracy values between two algorithms are statistically significant or not (signified by 'Yes' or 'No' respectively) with a p-value determined from the test. Based on the table, the accuracies of all versions of FLBNN are not significantly different from the accuracy values of KNN, WKNN, and NNv3. The accuracy values of the FLBNN algorithm when the displacement percentage is at 80% and 90% are also not significantly different to the accuracy of the PFL algorithm. However, FLBNN does achieve significantly improved accuracy than NNv4, NNv2, DFL, and SRL-KNN. In addition, the FLBNN algorithm with displacement percentage values between 10% and 70% is more accurate compared to PFL. Therefore, Hypothesis 3.1 is adequately verified when FLBNN (i.e., displacement percentage from 10% to 90%) is compared to NNv4, NNv2, DFL, SRL-KNN.

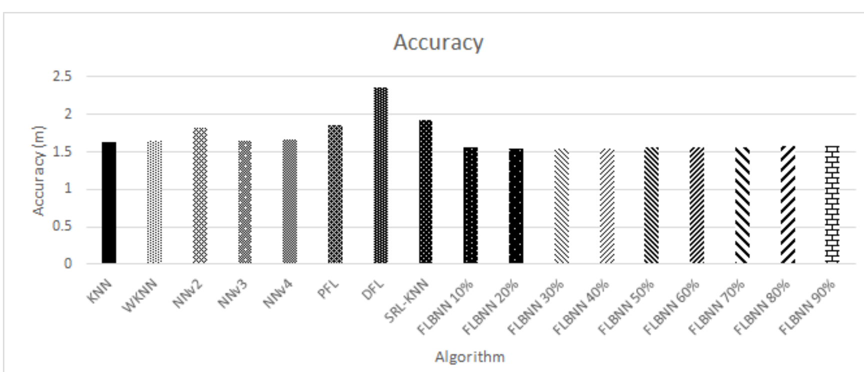

Figure 4-1: The Accuracies for The Fingerprinting-Based Algorithms.

In addition, the hypothesis is adequately verified when FLBNN (i.e., displacement percentage from 10% to 70%) is compared to PFL.

Table 4.1: The p-Values Related to The Accuracies

Algorithm	FLBNN 10%	FLBNN 20%	FLBNN 30%	FLBNN 40%	FLBNN 50%	FLBNN 60%	FLBNN 70%	FLBNN 80%	FLBNN 90%
KNN	0.438 (No)	0.438 (No)	0.438 (No)	0.438 (No)	0.438 (No)	0.438 (No)	0.438 (No)	0.438 (No)	0.438 (No)
WKNN	0.199 (No)	0.199 (No)	0.199 (No)	0.199 (No)	0.199 (No)	0.199 (No)	0.171 (No)	0.171 (No)	0.171 (No)
NNv2	0.011 (Yes)	0.004 (Yes)	0.004 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
NNv3	0.074 (No)	0.074 (No)	0.177 (No)	0.177 (No)	0.231 (No)	0.231 (No)	0.315 (No)	0.252 (No)	0.252 (No)
NNv4	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.002 (Yes)	0.002 (Yes)
PFL	0.008 (Yes)	0.008 (Yes)	0.008 (Yes)	0.008 (Yes)	0.008 (Yes)	0.008 (Yes)	0.008 (Yes)	0.072 (No)	0.072 (No)
DFL	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)
SRL-KNN	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)

The fingerprinting-based algorithms are also compared to FLBNN in terms of precision when all four APs are deployed (see figure 4-2). NNv3 achieves the highest precision compared to every other fingerprinting algorithm. The DFL algorithm also achieves the lowest precision. Each existing fingerprinting algorithm and FLBNN are statistically compared in terms of their precision by using the Levene's test for data related to standard deviations (see Table 4.2). The precision values of the FLBNN algorithm are not significantly different compared to NNv4 and SRL-KNN. Therefore, Hypothesis 3.2 is adequately verified when the algorithm (i.e., shifting percentage ranges from 10% to 90%) is compared to PFL and DFL. In addition, FLBNN achieves a similar precision to NNv4 and SRL-KNN.

Although FLBNN runs in O(t * p), where t is the number of deployed centroids and p is the number of every Path Loss exponent (PLe) values calculated using Receive

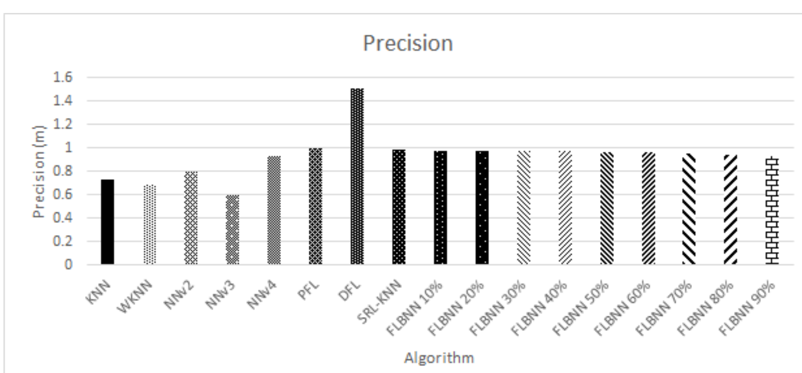

Figure 4-2: The Precision Values for The Fingerprinting-Based Algorithms.

Table 4.2: The p-Values Related to The Precision Values

Algorithm	FLBNN 10%	FLBNN 20%	FLBNN 30%	FLBNN 40%	FLBNN 50%	FLBNN 60%	FLBNN 70%	FLBNN 80%	FLBNN 90%
KNN	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
WKNN	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
NNv2	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
NNv3	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
NNv4	0.080 (No)	0.092 (No)	0.114 (No)	0.148 (No)	0.202 (No)	0.282 (No)	0.395 (No)	0.545 (No)	0.733 (No)
PFL	0.004 (Yes)	0.004 (Yes)	0.005 (Yes)	0.006 (Yes)	0.008 (Yes)	0.011 (Yes)	0.016 (Yes)	0.024 (Yes)	0.036 (Yes)
DFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
SRL-KNN	0.701 (No)	0.656 (No)	0.583 (No)	0.487 (No)	0.380 (No)	0.274 (No)	0.181 (No)	0.108 (No)	0.059 (No)

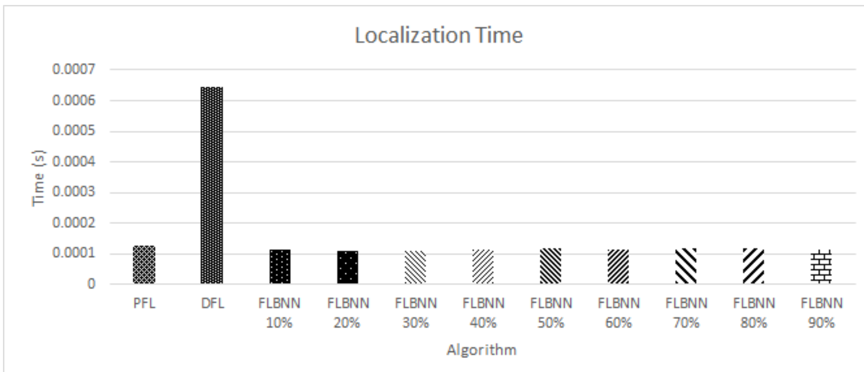

Figure 4-3: The Localization Time Values for The Fingerprinting-Based Algorithms.

Signal Strength Indicator (RSSI) measurements during the offline stage, the run times for PFL and DFL are unknown (see Table 2.1 from Chapter 2). Another goal from this experiment is to check the impact of the number of APs on the performance of the algorithms.

The PFL and DFL algorithms are compared to FLBNN in terms of run time when all four APs are used (see figure 4-3). Each algorithm is statistically compared based on their run time by using the Mann-Whitney U test for non-parametric data (see Table 4.3). The run time values of each algorithm are all significantly different from the run time values of each version of FLBNN. The FLBNN algorithm improves the run time metric compared to PFL and DFL.

Table 4.3: The p-Values Related to The Localization Time Values

Algorithm	FLBNN 10%	FLBNN 20%	FLBNN 30%	FLBNN 40%	FLBNN 50%	FLBNN 60%	FLBNN 70%	FLBNN 80%	FLBNN 90%
PFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
DFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)

4.2 Results Based on the Highest Correlated Access Points

Another set of results demonstrate the deployment of only two of the four APs. Essentially, selected combinations of the four APs are evaluated in terms of their accuracy in order to determine which AP combination achieves the highest correlation (see Table 4.4). Unlike the results demonstrated in Section 4.1, these results consider any statistical dependence between APs. There are five zones in total. The accuracy value here refers to the percentage of correctly classified zones when the target device is standing in each zone (see Table 4.5). In Table 4.4, APs 2 and 3 achieve the best average accuracy of 58.4% over 140 samples.

Table 4.4: Access Point Combination Accuracy Comparison

Algorithm	Average Accuracy (%)
APs 2 and 3	0.58381
APs 1, 2, and 4	0.389206
APs 1 and 2	0.389206
APs 1, 2, 3, and 4	0.389206
APs 3 and 4	0.289206
APs 1 and 3	0.194603
APs 2, 3, and 4	0.194603
APs 1, 3, and 4	0.194603
APs 1, 2, and 3	0.194603
APs 1 and 4	0.194603
APs 2 and 4	0

Table 4.5: Access Point 2 and 3 Confusion Matrix

		Actual Zone				
		Zone 1	Zone 2	Zone 3	Zone 4	Zone 5
Predicted Zone	Zone 1	0	0	0	0	0
	Zone 2	0.973016	0.973016	0	0.945238	0
	Zone 3	0	0	0.973016	0.028571	0
	Zone 4	0	0	0	0	0
	Zone 5	0	0	0	0	0.973016

*All the confusion matrices can be provided upon request.

Every fingerprinting-based algorithm is compared to FLBNN in terms of accuracy when only two of the four APs are used (see figure 4-4). FLBNN with a displacement percentage of 50% achieves a higher accuracy compared to NNv2, NNv4, PFL, and DFL. In addition, PFL achieves the lowest accuracy. Based on an analysis of the results (see Table 4.6), every accuracy value of each existing fingerprinting-based algorithm is significantly different compared to the accuracy values of every version of the FLBNN algorithm. Therefore, Hypothesis 3.1 is adequately verified when FLBNN (i.e., displacement percentage from 10% to 90%) is compared to NNv4, NNv2, PFL, and DFL.

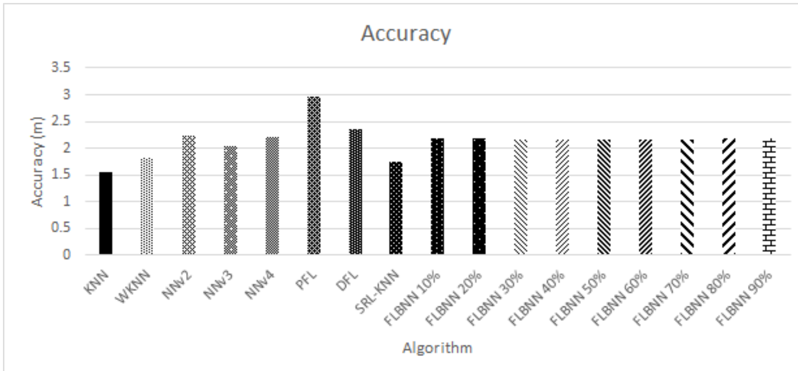

Figure 4-4: Fingerprinting Algorithm Accuracy Comparison for Selected Access Points.

Table 4.6: Fingerprinting Algorithm Accuracy Statistical Analysis for Best Access Point Selection

Algorithm	FLBNN 10%	FLBNN 20%	FLBNN 30%	FLBNN 40%	FLBNN 50%	FLBNN 60%	FLBNN 70%	FLBNN 80%	FLBNN 90%
KNN	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
WKNN	0.000 (Yes)	0.000 (Yes)	0.017 (Yes)	0.017 (Yes)	0.017 (Yes)	0.017 (Yes)	0.017 (Yes)	0.011 (Yes)	0.011 (Yes)
NNv2	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)
NNv3	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)
NNv4	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
PFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
DFL	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)	0.034 (Yes)
SRLKNN	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.001 (Yes)	0.001 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)

While using only two of the four APs, the existing fingerprinting-based algorithms are compared to FLBNN in terms of precision (see figure 4-5). The NNv3 algorithm achives the highest precision while the DFL algorithm achieves the lowest precision. Based on an analysis of the results (see Table 4.7), the precision values of the FLBNN algorithm are not statistically significant when compared to NNv4 and SRL-KNN. However, the precision values of FLBNN are significantly different compared to every other fingerprinting-based algorithm. Therefore, Hypothesis 3.2 is adequately verified when FLBNN is compared to both PFL and DFL and as precise as the NNv4 and SRL-KNN algorithms.

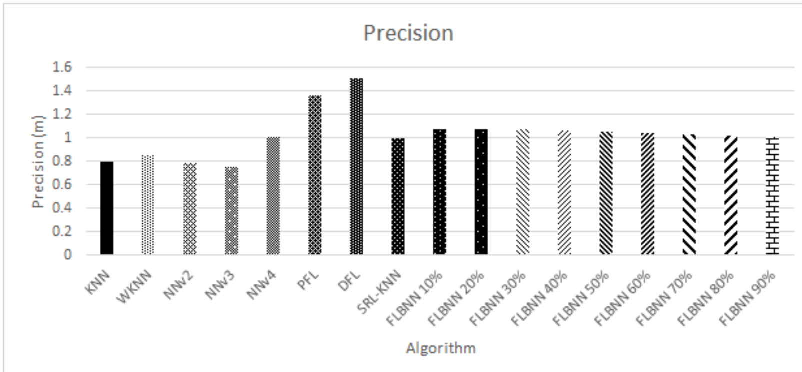

Figure 4-5: Fingerprinting Algorithm Precision Comparison for Selected Access Points.

Table 4.7: Fingerprinting Algorithm Precision Statistical Analysis for Best Access Point Selection

Algorithm	FLBNN 10%	FLBNN 20%	FLBNN 30%	FLBNN 40%	FLBNN 50%	FLBNN 60%	FLBNN 70%	FLBNN 80%	FLBNN 90%
KNN	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.001 (Yes)	0.002 (Yes)	0.003 (Yes)	0.004 (Yes)	0.007 (Yes)	0.013 (Yes)
WKNN	0.004 (Yes)	0.004 (Yes)	0.004 (Yes)	0.006 (Yes)	0.008 (Yes)	0.011 (Yes)	0.016 (Yes)	0.025 (Yes)	0.039 (Yes)
NNv2	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
NNv3	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.001 (Yes)	0.001 (Yes)
NNv4	0.421 (No)	0.437 (No)	0.469 (No)	0.519 (No)	0.589 (No)	0.679 (No)	0.790 (No)	0.919 (No)	0.938 (No)
PFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
DFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
SRLKNN	0.165 (No)	0.172 (No)	0.189 (No)	0.215 (No)	0.254 (No)	0.308 (No)	0.379 (No)	0.469 (No)	0.580 (No)

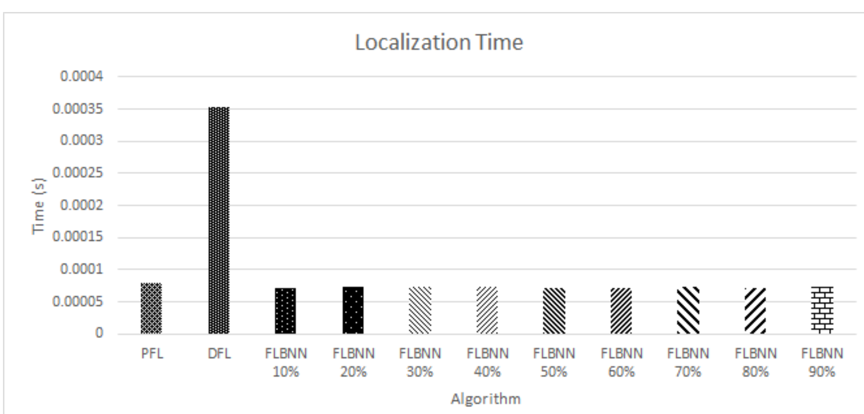

Figure 4-6: Fingerprinting Algorithm Localization Time Comparison for Selected Access Points.

Finally, the PFL and DFL algorithms are compared to FLBNN in terms of run time using only two APs (see figure 4-6). The run time values of each algorithm are all significantly different from the run time values of FLBNN (see Table 4.8). Thus, the FLBNN algorithm has a faster run time compared to PFL and DFL.

Table 4.8: Fingerprinting Algorithm Localization Time Statistical Analysis for Best Access Point Selection

Algorithm	FLBNN 10%	FLBNN 20%	FLBNN 30%	FLBNN 40%	FLBNN 50%	FLBNN 60%	FLBNN 70%	FLBNN 80%	FLBNN 90%
PFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)
DFL	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)	0.000 (Yes)

Chapter 5

Discussion

In this chapter, we mention the possible explanations behind the results demonstrated in Chapter 4 in terms of the accuracy, precision, and run time metrics. In addition, we highlight the added values to the existing literature in the area of localization. Furthermore, we mention the limitations and recommendations for future work.

5.1 The Deployment of All Access Points

The first major finding in this research is that our Fingerprinting Line-Based Nearest Neighbor algorithm (FLBNN) achieves higher accuracy compared to those for all the existing algorithms when all four Access Points (APs) are deployed. Our explanation for these results follows Hypothesis 3.1 outlined in Chapter 3. To reiterate, FLBNN has a higher accuracy compared to those for the Nearest Neighbor-based algorithms, because the algorithm possibly shifts the Midpoint M closer to the target node, and therefore, the computed centroid is closer to the actual location.

A more interesting observation to mention is that the FLBNN algorithm also achieves higher precision compared to those for Path-Loss-Based Fingerprint Localization (PFL), Dual-Scanned Fingerprint Localization (DFL), and is just as precise as Nearest Neighbor version 4 (NNv4) and Soft-Range-Limited K-Nearest Neighbor (SRL-KNN). Our explanation for why FLBNN achieves a greater precision follows Hypothesis 3.2. In other words, FLBNN has a higher precision because the algorithm calculates the estimated locations within a particular area of the space (i.e., triangle). This explanation also assists to demonstrate why NNv4 achieves a similar precision to that for FLBNN. The reason as to why FLBNN achieves a higher precision compared to those for PFL and DFL is because in PFL, the algorithm chooses the nearest grid point. In comparison, FLBNN chooses the nearest centroid and then chooses the two closest grid points. A small change in the Receive Signal Strength Indicator (RSSI) measurements can significantly impact PFL, in which a small change causes PFL to choose a new closest grid point that is 2.5 m away from the original. In comparison, FLBNN chooses the nearest centroid. As a result, the centroids have more distinctive feature values compared to grid points with all four APs, and as a result, FLBNN is not susceptible to changes in RSSIs. In addition, DFL achieves the lowest precision because the RSSI and distance thresholds only include the grid points that are within the center of the grid point space. As a result, based on the RSSI measurements that DFL reads, it cannot detect locations that are outside of the range of the thresholds. This is usually the case for grid points that are at the far edges of the grid point space. Therefore, DFL achieves the lowest precision. Moreover, SRL-KNN achieves a higher precision compared to that for DFL. The reason for this behavior is that even though SRL-KNN uses the previously calculated location as a threshold for choosing grid points, this threshold is not calculated within the center of the room compared to DFL.

In terms of run time, the FLBNN algorithm has a faster run time compared to those for PFL and DFL. As mentioned in the previous chapter, the FLBNN algorithm runs in O(t * p) where t is the number of deployed centroids and p is the number of Path Loss exponent (PLe) values. However, the run times for PFL and DFL are unknown (see Table 2.1 in Chapter 2). Therefore, the run time of the FLBNN algorithm was compared to PFL and DFL using the wall clock time. The results demonstrate that the FLBNN algorithm runs faster compared to those for the existing algorithms. This is because FLBNN initially searches for the nearest centroid in the grid point space, and then evaluates the four surrounding grid points to determine the two closest grid points. In comparison, PFL evaluates every grid point in the space to determine the closest grid point as the localized position. Additionally, DFL has a significantly slower run time compared to those for both FLBNN and PFL because DFL focuses on the calculation of an RSSI threshold and a distance threshold. These two thresholds only include the grid points in the space to be chosen as the localized point before the calculation of the nearest grid point.

For precision, the K-Nearest Neighbor (KNN), Weighted KNN (WKNN), Nearest Neighbor version 2 (NNv2), and Nearest Neighbor version 3 (NNv3) algorithms achieve a higher precision compared to that for FLBNN. The reason as to why KNN and consequently WKNN achieves the highest precision is because KNN chooses the four closest grid points compared to that for FLBNN. The FLBNN algorithm chooses the nearest centroid and then chooses the two closest grid points of the four surrounding ones. With KNN, changes in RSSIs do not affect the algorithm as much because the four closest grid points remain the same. The only difference is that one of those four grid points might be closer to the target than another in a different run of the algorithm. However, in all runs of KNN, the four closest grid points remain the same, even though the closest grid point may change. In comparison, FLBNN chooses the nearest centroid and then the two closest grid points. Even though the nearest centroid does not change during several runs of FLBNN, a fluctuation in RSSI measurements alters the choice of the two closest grid points. Thus, FLBNN achieves a lower precision in comparison to KNN and WKNN. For NNv2, it achieves a higher precision because it chooses the nearest centroid instead of additionally choosing the two closest grid points. The change in RSSIs modifies the choice of the two nearest grid points, and as a result, the precision of FLBNN is lowered compared to that for NNv2. Finally, NNv3 achieves a higher precision compared to that for FLBNN. Although the NNv3 algorithm relies on choosing two closest grid points after choosing the four closest grid points, NNv3 can choose the two closest grid points where the line between the points can intersect Centroid S. In comparison, FLBNN must choose the two closest grid points where the line drawn between these points does not cross Centroid S. As a result, when NNv3 changes its two closest grid points during consecutive runs, the localized point does not change as much. However, the localized

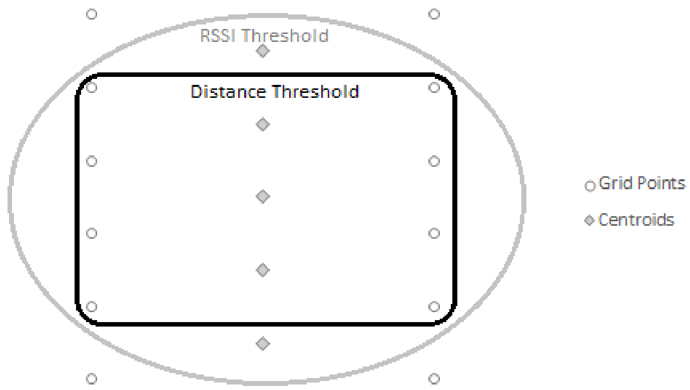

Figure 5-1: Dual-Scanned Fingerprint Localization Thresholds.

points computed by FLBNN are farther away from each other in consecutive runs because a change in one of the closest grid points changes the location of the triangle where the localized point is calculated.

There are a couple of interesting observations beyond FLBNN's performance. An interesting observation is that DFL achieves lower accuracy compared to that for PFL. The key reason for this is because the RSSI and distance thresholds filter the grid points that are within the center of the grid point space (see figure 5-1). These thresholds are often not calculated around the edges of the grid point space. As a result, the DFL algorithm cannot accurately predict the location of a user that is around the edges of a grid point space because the thresholds are calculated in the center of the grid point space.

NNv3 achieves higher precision compared to that for NNv2, even though NNv2 chooses the nearest centroid only. This is hypothesized that a change in the closest centroid during several runs of NNv2 can significantly impact the distance between localized points. The distance between localized points are not influenced as much in NNv3. If a change in RSSI measurements appears, the two closest grid points may change. However, because the four closest grid points are often close together, a change in the two closest grid points does not impact the distance between the localized points. However, a change in the nearest centroid in NNv2 results in a distance of 2.5 m between localized points. Therefore, NNv3 has higher precision compared to that for NNv2.

WKNN achieves a higher precision compared to that for KNN when all four APs are deployed. This is because even though WKNN achieves a lower accuracy, the weighted average calculation uses the distance between the grid point and the target. Even though the localized position is not close to the target, the weighted average mostly calculates the points to be closer to each other. In comparison, KNN calculates the centroid between the four closest grid points without a weighted average. Thus, KNN calculates the points to be farther away from each other and WKNN achieves a higher precision in comparison.

Finally, when discussing WKNN, SRL-KNN, which is based on WKNN, achieves a lower precision compared to that for WKNN. The reason for this is because SRL-KNN chooses its four closest grid points based on a threshold surrounding the previous localized point calculated. The threshold limits the choice of the four closest grid points, and thus, a change in RSSI measurements significantly impacts the change in the four closest grid points. In comparison, the four closest grid points in WKNN do not change over consecutive runs compared to SRL-KNN because there are more grid points to choose from when choosing the nearest four grid points in WKNN. Therefore, SRL-KNN achieves a lower precision compared to WKNN.

5.2 Highest Correlated Access Points

The next major finding in this research is that FLBNN achieves a higher accuracy compared to those for NNv2, NNv4, PFL, and DFL when only a chosen combination of APs are used. Once again, our explanation for this result follows Hypothesis 3.1 outlined in Chapter 3. In addition, the FLBNN algorithm also achieves a greater precision than PFL, greater precision than DFL, and is just as precise as NNv4 and SRL-KNN. The explanation of this behavior follows the same explanation when all four APs are deployed. As explained previously, when two of the four APs are used, FLBNN also achieves a faster run time than PFL and DFL.

However, the KNN, WKNN, NNv3, and SRL-KNN algorithms have higher accuracy than FLBNN when only two APs are used. This is primarily because the methodologies of these four algorithms involve calculating the four nearest grid points in comparison to FLBNN. When only two APs are used, the fingerprints for each grid point are reduced. As a result, some points may be more indistinguishable in the grid point space when finding the nearest grid points. This is especially the case with the centroids in the center of the room. However, the fingerprints of each grid point are more distinguishable from each other. As a result, the initial centroid choice is inaccurate compared to choosing four grid points. Therefore, the FLBNN algorithm achieves a lower accuracy compared to those algorithms when only two of the four APs are used.

There are various other observations beyond FLBNN's performance when two APs are used. First, DFL achieves a better accuracy compared to PFL when only two APs are used. This is primarily because DFL is not affected as much by the AP deployment. DFL still calculates the thresholds to be within the center of the room. However, because the fingerprints of each grid point are limited, there are some grid points that are similar to each other as a result. In the case of PFL, the fingerprints between two grid points are less distinguishable than before when the four APs are deployed. As a result, PFL chooses a different nearest grid point during different runs of the algorithm. In comparison, because DFL only chooses grid points within the thresholds calculated, some grid points with similar fingerprints are excluded from these thresholds. As a result, DFL is not affected as much by the change in the fingerprints of grid points. Therefore, DFL achieves a higher accuracy than PFL.

WKNN achieves a lower precision compared to KNN when only two APs are used even though the four closest grid points do not change for both KNN and WKNN in consecutive runs of the algorithms. With the updated distance values, WKNN's localized points can be possibly more varied compared to those for KNN because of the weighted average calculation. A change in distance values between the target node and a grid point in KNN does not significantly impact the calculation of the localized point. Given that the distance values are less distinct due to less distinct fingerprints on grid points, those distance values are used in WKNN when the weighted average between the chosen grid points is calculated. As a result, a change in RSSI values can affect the localized points in WKNN more than in KNN, and thus, WKNN has a lower precision.

SRL-KNN achieves a higher accuracy than WKNN when only two APs are used. Initially, WKNN outperformed SRL-KNN with all four APs. To reiterate, SRL-KNN has a low precision when all four APs are used due to the impact of the RSSIs changing the four closest grid points in the threshold. However, within the threshold created by the previously calculated location, the SRL-KNN algorithm is not impacted as much by the change in RSSI values. As a result, SRL-KNN cannot localize points that are more accurate over time, and therefore, it does not achieve the accuracy of WKNN when all four APs are used. However, when two APs are used, SRL-KNN achieves a higher accuracy. This is because the initial previous location calculated when only two APs are used for both WKNN and SRL-KNN is accurate. As a result, a fluctuation in RSSI values impacts WKNN more than SRL-KNN because the localized points calculated from WKNN can be possibly more inaccurate in later runs of the algorithm in a given test. However, SRL-KNN is not affected as much by the change in RSSI over the entire grid point space because of the threshold set by the previous localized points. Therefore, SRL-KNN achieves a higher accuracy than WKNN when only two APs are used.

SRL-KNN achieves a higher accuracy than PFL when only two APs are used. Originally, SRL-KNN had a lower accuracy than PFL when all four APs are used. PFL chooses the closest grid point compared to SRL-KNN, which chooses the four closest grid points. As a result, when certain fingerprinting features are removed from the grid points, there were some grid points that PFL chose that are far away from the target node. However, SRL-KNN is not impacted as much because a change in one of the closest four grid points does not affect the localized point significantly. As a result, SRL-KNN achieves a higher accuracy when only two APs are used.

Compared to NNv3, SRL-KNN achieves a higher accuracy than the NNv3 algorithm. The reason for this is because similar to PFL, the choice of the two closest grid points in NNv3 is significantly affected. Even though both algorithms first choose the four closest grid points and calculate a centroid between them, NNv3 chooses the two closest grid points in later steps. When fingerprinting features were removed from grid points, the NNv3 algorithm chooses two different closest grid points as a result. Therefore, the SRL-KNN algorithm achieves a higher accuracy than NNv3.

Finally, SRL-KNN achieves a higher accuracy than NNv4 and NNv2 when only two APs are used. The explanation for this behavior is similar to the explanation for why SRL-KNN has higher accuracy than FLBNN when only two APs are used. Since NNv2 and NNv4 both involve choosing the nearest centroid first, the initial centroid choice can be possibly inaccurate compared to the actual closest centroid to the target node. Since SRL-KNN chooses the four closest grid points, however, the SRL-KNN algorithm achieves a higher accuracy.

5.3 Limitations and Recommendations

One of the key limitations in this study is that the grid point space used is relatively small and only a few centroids exist within the space. However, the grid point space is designed in such a way to follow the Cisco best practices for AP spacing, where the grid point space should ideally be designed within the convex boundary formed by all four APs [45]. With regards to future study of the FLBNN algorithm and fingerprinting-based algorithms, one recommendation is to work with a larger grid point space to further explore the performance of each algorithm within various different spaces. A major limitation in this study is that only four test points were used during the experiments. Therefore, there is a need to deploy more test points in future experiments in order to further evaluate the performance of each algorithm. Another limitation is that there is a small set of APs that were used in the experiments. More APs are recommended to be deployed in future experiments in order to explore how the number of APs influences the performance of each algorithm. We have not explored the challenges when varying Wi-Fi environment settings such as channel selection, radio frequency selection, and physical ambience. Thus, there is a need to investigate the accuracy and precision of the FLBNN algorithm based on the mentioned settings. Finally, we have only experimented with Wi-Fi technologies and have not explored other technologies such as Bluetooth and Zigbee. Thus, there is a need to evaluate each algorithm with those technologies.

Chapter 6

Conclusion

In conclusion, this Thesis answers the question of whether the line-based shifting of the grid points improves the indoor localization metrics compared to those for several existing fingerprinting-based algorithms [17] [21] [22] [24]. The Thesis focuses on important metrics such accuracy, precision, and localization time. A thorough investigation was conducted in order to verify the hypotheses. Furthermore, a detailed discussion was mentioned in order explain the different behaviors for the Nearest Neighbor-based algorithms in terms of all those metrics. In this chapter, we summarize the conducted research and mention some future directions.

6.1 Summary

Indoor localization is a growing research area considering that there are fields such as healthcare that is positively impacted by this area. Technologies such as GPS are used to determine the location of a user outdoors, but is not sufficient to localize a user indoors [6]. Moreover, techniques such as Angle of Arrival (AoA), Time of Arrival (ToA) and Time Difference of Arrival (TDoA) are not sufficient for locating a user indoors [2]. In comparison to the non-Receive Signal Strength Indicator (RSSI) techniques, trilateration is an RSSI-based technique, but requires additional infrastructure and is not effective in spatially diverse environments [9] [10]. In order to address these issues, fingerprinting is not as susceptible to angle multipath and can achieve higher accuracy without an Analog to Digital Converter (ADC) and clock synchronization [11] [12] [14] [18]. Additionally, the fingerprinting technique is more applicable for spatially diverse indoor environments [9] [10] [17].

Fingerprinting algorithms such as the Nearest Neighbor (NN) and K-Nearest Neighbor (KNN) suffer from multipath interference [17]. Weighted KNN (WKNN) has high time complexity [20] [22]. WKNN, Path-Loss-Based Fingerprint Localization (PFL), and Dual-Scanned Fingerprint Localization (DFL) have relatively low accuracy [20] [22] [24]. Additionally, the time complexity of PFL and DFL were not evaluated in their research [24]. Moreover, Nearest Neighbor version 2 (NNv2), Nearest Neighbor version 3 (NNv3), and Nearest Neighbor version 4 (NNv4) all achieve high accuracy, however, the accuracy and precision metrics can be further improved [17]. For Soft-Range-Limited KNN (SRL-KNN), it has potential inaccuracies in locations calculated in previous runs of the algorithm and its precision is unknown [21]. Therefore, this Thesis proposed and implements the Fingerprinting Line-Based Nearest Neighbor (FLBNN) algorithm in order to improve the accuracy, precision, and run time metrics.

Our findings indicate that while deploying all Access Points (APs), our proposed FLBNN algorithm improves accuracy compared to those for NNv4, NNv2, DFL, and SRL-KNN. FLBNN also improves accuracy compared to those for NNv2, NNv4, PFL, and DFL when only two APs are deployed. This is primarily because the new algorithm has the shifting capability.

When all the four APs are deployed, FLBNN also achieves higher precision compared to those for PFL and DFL and achieves similar precision compared to those for NNv4 and SRL-KNN. This is hypothesized that the localized positions are within a particular area of the grid space (i.e., triangle). In addition, when only a subset of APs is deployed, FLBNN does not achieve a higher precision compared to those for KNN, WKNN, NNv2, and NNv3 for the same reason as when all four APs are deployed.

Finally, our proposed FLBNN algorithm improves the run time compared to those for PFL and DFL based on the full and partial deployments of APs. This is because

PFL focuses on evaluating all grid points to locate the nearest grid point and DFL relies on calculating two thresholds. The key contributions of this work are listed as follows.

1. The FLBNN algorithm improves accuracy compared to those for NNv4, NNv2, DFL, SRL-KNN when all four APs are used. FLBNN also improves accuracy compared to those for NNv2, NNv4, PFL, and DFL algorithms when a subset of APs are deployed.

2. The FLBNN algorithm improves precision compared to those for PFL and DFL when all four APs or a subset of APs are deployed.

3. The FLBNN algorithm improves the run time complexity compared to those for PFL and DFL.

6.2 Future Work

As future work, we will explore the deployment of Bluetooth beacons for our FLBNN algorithm along with using a larger grid point space. In addition, more test points and APs will be deployed in order to further evaluate the performance of each existing fingerprinting-based algorithm and the FLBNN algorithm. Various different techniques beyond the Kalman filter will be implemented to filter the RSSI measurements [16]. Moreover, the proposed and existing algorithms will be extended to 3-D. Furthermore, various machine learning techniques will be applied to these algorithms in order to investigate the performance of each algorithm. The accuracy and precision of the FLBNN algorithm when changing several environmental settings such as Dynamic Frequency Selection (DFS) and channel selection will also be explored. Finally, new variations of the FLBNN algorithm will be proposed in the future.

Bibliography

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[2] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018.

[3] R.-S. Hsiao, C.-H. Kao, T.-X. Chen, and J.-L. Chen, "A passive RFID-based location system for personnel and asset monitoring," *Technology and Health Care*, vol. 26, no. 1, pp. 11–16, Feb. 2018.

[4] M. Zvěřová, "Clinical aspects of Alzheimers disease," *Clinical Biochemistry*, vol. 72, pp. 3–6, 2019.

[5] Ó. Belmonte-Fernández, A. Puertas-Cabedo, J. Torres-Sospedra, R. Montoliu-Colás, and S. Trilles-Oliver, "An Indoor Positioning System Based on Wearables for Ambient-Assisted Living," *Sensors*, vol. 17, no. 12, p. 36, 2016.

[6] F. Dalkilic, U. C. Cabuk, E. Arikan, and A. Gurkan, "An analysis of the positioning accuracy of iBeacon technology in indoor environments," *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017.

[7] Y. Wang, M. Li, L. Bai, T. Fu, and F. Gao, "Normalized amplitude of channel state information: The robust parameter for indoor localization in wireless sensor

networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 2, p. 155014771875942, 2018.

[8] F. Liu, J. Liu, Y. Yin, W. Wang, D. Hu, P. Chen, and Q. Niu, "Survey on WiFi-based indoor positioning techniques," *IET Communications*, vol. 14, no. 9, pp. 1372–1383, 2020.

[9] X. Zhou, T. Chen, D. Guo, X. Teng, and B. Yuan, "From one to crowd: a survey on crowdsourcing-based wireless indoor localization," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 423–450, Nov. 2018.

[10] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor Fingerprint Positioning Based on Wi-Fi: An Overview," *ISPRS International Journal of Geo-Information*, vol. 6, no. 5, p. 135, 2017.

[11] T. Xie, H. Jiang, X. Zhao, and C. Zhang, "A Wi-Fi-Based Wireless Indoor Position Sensing System with Multipath Interference Mitigation," *Sensors*, vol. 19, no. 18, p. 3983, 2019.

[12] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2017.

[13] J. Xu, C. Law, and M. Ma, "Performance of time-difference-of-arrival ultra wide-band indoor localisation," *IET Science, Measurement & Technology*, vol. 5, no. 2, pp. 46–53, Jan. 2011.

[14] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys*, vol. 46, no. 2, pp. 1–32, Jan. 2013.

[15] M. Passafiume, S. Maddio, and A. Cidronali, "An Improved Approach for RSSI-Based only Calibration-Free Real-Time Indoor Localization on IEEE 802.11 and 802.15.4 Wireless Networks," *Sensors*, vol. 17, no. 4, p. 717, 2017.

[16] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, "Indoor Positioning Algorithm Based on the Improved RSSI Distance Model," *Sensors*, vol. 18, no. 9, p. 2820, 2018.

[17] T. E. Salti, M. Orlando, S. Hood, G. Knelsen, M. Iarocci, Z. Lazzara, Y. Xie, J. C.-C. Cheung, and I. Woungang, "A New Set of Bluetooth-Based Fingerprinting Algorithms for Indoor Location Services," *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2018.

[18] A. F. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, "Localization and Positioning Systems for Emergency Responders: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2836–2870, 2017.

[19] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarrone, "Smart probabilistic fingerprinting for WiFi-based indoor positioning with mobile devices," *Pervasive and Mobile Computing*, vol. 31, pp. 107–123, 2016.

[20] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064).*

[21] M. T. Hoang, Y. Zhu, B. Yuen, T. Reese, X. Dong, T. Lu, R. Westendorp, and M. Xie, "A Soft Range Limited K-Nearest Neighbors Algorithm for Indoor Localization Enhancement," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10208–10216, 2018.

[22] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825–845, 2005.

[23] R. Ma, Q. Guo, C. Hu, and J. Xue, "An Improved WiFi Indoor Positioning Algorithm by Weighted Fusion," *Sensors*, vol. 15, no. 9, pp. 21824–21843, 2015.

[24] J. Zhang, G. Han, N. Sun, and L. Shu, “Path-Loss-Based Fingerprint Localization Approach for Location-Based Services in Indoor Environments,” *IEEE Access*, vol. 5, pp. 13756–13769, 2017.

[25] T. Wang, Y. Liang, Y. Mei, M. Arif, and C. Zhu, “High-accuracy localization for indoor group users based on extended Kalman filter,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, p. 155014771881272, 2018.

[26] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms for wireless sensor networks: A survey,” *Information Fusion*, vol. 49, pp. 1–25, 2019.

[27] H. Tong, Q. Deng, T. Zhang, and Y. Bi, “A low-cost indoor localization system based on received signal strength indicator by modifying trilateration for harsh environments,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 6, p. 155014771877968, 2018.

[28] J. Yang, Y. Li, and W. Cheng, “An improved geometric algorithm for indoor localization,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 3, p. 155014771876737, 2018.

[29] U. M. Qureshi, Z. Umair, and G. P. Hancke, “Evaluating the Implications of Varying Bluetooth Low Energy (BLE) Transmission Power Levels on Wireless Indoor Localization Accuracy and Precision,” *Sensors*, vol. 19, no. 15, p. 3282, 2019.

[30] W. Liu, X. Luo, Q. Mu, Y. Liu, and F. Liu, “An indoor Wi-Fi access points localization algorithm based on improved path loss model parameter calculation method and recursive partition,” *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, p. 155014771985203, 2019.

[31] M. M. D. Horno, I. García-Varea, and L. O. Barbosa, “Calibration of Wi-Fi-Based Indoor Tracking Systems for Android-Based Smartphones,” *Remote Sensing*, vol. 11, no. 9, p. 1072, Jun. 2019.

[32] K. Huang, K. He, and X. Du, “A Hybrid Method to Improve the BLE-Based Indoor Positioning in a Dense Bluetooth Environment,” *Sensors*, vol. 19, no. 2, p. 424, 2019.

[33] N. Bnilam, E. Tanghe, J. Steckel, W. Joseph, and M. Weyn, “ANGLE: ANGular Location Estimation Algorithms,” *IEEE Access*, vol. 8, pp. 14620–14629, 2020.

[34] Y. Zheng, M. Sheng, J. Liu, and J. Li, “Exploiting AoA Estimation Accuracy for Indoor Localization: A Weighted AoA-Based Approach,” *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 65–68, 2019.

[35] D. Liu, Y. Wang, P. He, Y. Zhai, and H. Wang, “TOA localization for multipath and NLOS environment with virtual stations,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, Sep. 2017.

[36] I. Guvenc and C.-C. Chong, “A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[37] A. Ferreira, D. Fernandes, A. Catarino, and J. Monteiro, “Performance Analysis of ToA-Based Positioning Algorithms for Static and Dynamic Targets with Low Ranging Measurements,” *Sensors*, vol. 17, no. 8, p. 1915, 2017.

[38] T. Lei, S. Pan, K. Wang, and Y. Yu, “Continuous TOA Measurement Based Single-Anchor Indoor Localization,” *IEEE Access*, pp. 1–1, 2019.

[39] S. He and S.-H. G. Chan, “Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[40] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, “An Improved K-Nearest-Neighbor Indoor Localization Method Based on Spearman Distance,” *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 351–355, March 2016.

[41] F. Yu, M. Jiang, J. Liang, X. Qin, M. Hu, T. Peng, and X. Hu, "5G WiFi Signal-Based Indoor Localization System Using Cluster k-Nearest Neighbor Algorithm," *International Journal of Distributed Sensor Networks*, vol. 10, no. 12, p. 247525, 2014.

[42] D. B. Ninh, J. He, V. T. Trung, and D. P. Huy, "An effective random statistical method for Indoor Positioning System using WiFi fingerprinting," *Future Generation Computer Systems*, vol. 109, pp. 238–248, 2020.

[43] L. Zhang, T. Du, and C. Jiang, "Detection of an unknown radio transmitter using an enhanced K-nearest neighbor algorithm based on virtual reference point and RSSD information," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[44] Y. Shi, W. Zhang, Z. Yao, M. Li, Z. Liang, Z. Cao, H. Zhang, and Q. Huang, "Design of a Hybrid Indoor Location System Based on Multi-Sensor Fusion for Robot Navigation," *Sensors*, vol. 18, no. 10, p. 3581, 2018.

[45] "Wi-Fi Location-Based Services 4.1 Design Guide - Best Practices-Location-Aware WLAN Design Considerations [Design Zone for Mobility]," *Cisco*, 30-Jan-2014. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG/wifich5.html. [Accessed: 25-Mar-2020].

Glossary

Access Point (AP) A device that emits Wi-Fi signals.

Accuracy A metric referring to the distance error between the location estimated and the actual location.

Angle of Arrival (AoA) A measurement that indicates the angle of a signal's arrival at a signal receiver sent from the target device [12].

Fingerprinting A technique which involves constructing a radio map with RSSI measurements and using it to determine the target device's location indoors [17].

Precision A metric referring to the distribution of the estimated locations and the relative distances between them.

Receive Signal Strength Indicator (RSSI) A measurement that indicates the strength of a signal that has been sent from an access point to the target device.

Time Complexity The run time complexity of an algorithm. A lower time complexity means that the algorithm runs faster.

Time Difference of Arrival (TDoA) A measurement that indicates the difference in signal arrival time between a pair of signal receivers [2] [13].

Time of Arrival (ToA) A measurement that indicates the time it takes for a signal to be transmitted from a signal transmitter to the target device [12].

Trilateration A technique which involves the use of three or more access points to send signals to the target device [10].