HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

An evaluation of passive crowdsourcing methods for large-scale construction of WiFi radio map

NGUYEN DUY HUNG

Hung.ND211267M@sis.hust.edu.vn

School of Information and Communication Technology

Supervisor: Associate Professor Ngo Quynh Thu

____

Supervisor's signature

School: Information and Communication Technology

HANOI, 10/2023

Graduation Thesis Assignment

Name: Nguyen Duy Hung

Phone: +84965476730

Email: Hung.ND211267M@sis.hust.edu.vn; hungnd1235@gmail.com

Class: 21A-IT-KHDL-E

Affiliation: Hanoi University of Science and Technology

Nguyen Duy Hung – hereby declare that this thesis, titled "*An evaluation of passive crowdsourcing methods for large-scale construction of WiFi radio map*" is my personal work, which is performed under the supervision of Associate Professor Ngo Quynh Thu. All data used for analysis in the thesis are my own research, analysis objectively and honestly, with a clear origin, and have not been published in any form. I take full responsibility for any dishonesty in the information used in this study.

Supervisor	Student
Signature and Name	Signature and Name

Acknowledgement

I would like to express my sincere gratitude to the people who have played an important role in the completion of my master's thesis.

First and foremost, I extend my deepest appreciation to my esteemed supervisor, Associate Professor Ngo Quynh Thu, whose guidance, expertise, and unwavering support have been instrumental in shaping this thesis. Your mentorship and valuable insights have significantly contributed to my academic journey.

To my beloved family, I am profoundly thankful for your continuous encouragement and sacrifices.

I would like to express my gratitude to Mr. Dinh Xuan Thuc and my teammates at Viettel Networks for creating the conditions that allowed me to dedicate the time and effort necessary to complete this research.

Finally, I wish to convey my heartfelt appreciation to my cherished lover. Your love, patience, and understanding have provided solace during the demanding phases of my academic pursuits.

This thesis is a culmination of the collective support and contributions from these exceptional individuals. I am deeply grateful!

Abstract

Indoor localization refers to the process of determining the location of a device or user within an indoor environment. In recent years, WiFi fingerprinting has become a popular approach for indoor localization. One of the key challenges of WiFi fingerprinting is the construction of a WiFi radio map, which involves creating a training dataset with WiFi measurements at reference locations within the area of interest. Passive crowdsourcing has been proposed to reduce the effort required for creating a WiFi radio map, enabling data collection without active user participation. Despite the promise of these methods, their testing and evaluation have been primarily limited to laboratory settings, creating a gap in their real-world application.

In this study, we leverage a large-scale indoor localization dataset, specifically the Microsoft Indoor Location 2.0, to evaluate the performance of passive crowdsourcing methods for large-scale WiFi radio map construction. In our experiments, we implement and test the highly influential Zee passive crowdsourcing approach along with various motion models. The results suggest that Zee's performance is promising for WiFi radio map construction when prior knowledge about the environment is available. Additionally, a comparative study on motion models indicates that a combination of auto-correlation-based step detection and the Weinberg stride length model yields the lowest absolute percentage error in traveled distance.

Student
Signature and Name

Table of Contents

List of Tables

List of Figures

Acronyms

Notation	Description
IoT	Internet of Things
WiFi	Wireless Fidelity, commonly used for wireless local area networking
RSSI	Received Signal Strength Indicator
AP	Access Point
ML	Machine Learning
RP	Reference Point
IMU	Inertial Measurement Unit
API	Application Programming Interface
Zee	Zero-effort Crowdsourcing
PIME	Placement Independent Motion Estimator
APF	Augmented Particle Filter
PF	Particle Filter
NASC	Normalized Auto-correlation based Step Counting
ZUPT	Zero Velocity Update
GB	Giga Byte
SSD	Solid State Drive
MDS	Multi-Dimensional Scaling
RSS	Received Signal Strength
ACF	Auto-correlation Function
MAPE	Mean Absolute Percentage Error
k-NN	k Nearest Neighbor

Chapter 1: Introduction

1.1 Research Background

The topic of indoor localization has received significant attention from both the academic world and industry in the last few decades [1] [2] [3] [4] [5]. Indoor localization refers to the process of obtaining a device or user location in an indoor setting or environment [1]. Over the past decade, the proliferation of smartphones and wearable devices with wireless communication capabilities have made the localization and tracking of such devices equivalent to the localization and tracking of corresponding users. As a result, it enabled a wide range of related applications and services, such as health care, industry, disaster management, building management, surveillance, Internet of Things (IoT) and smart architectures [1] [2] [4] [5] [6].

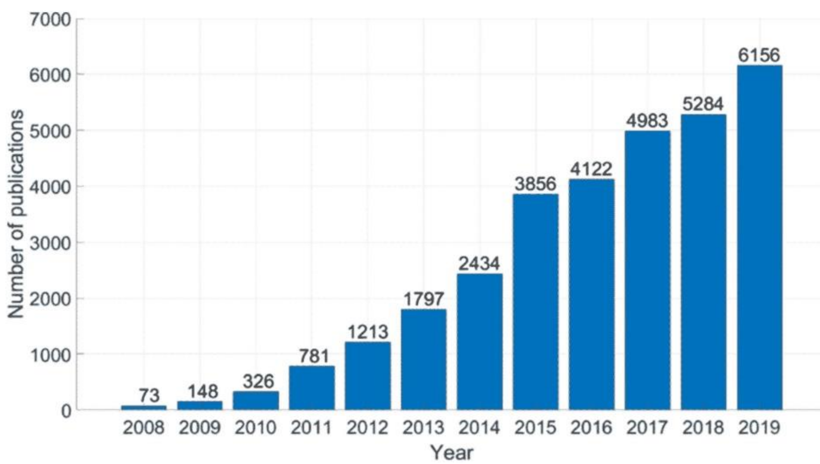

Figure 1 The number of publications on smartphones indoor localization and tracking indexed by Google Scholar [2]

Over the past few decades, numerous localization techniques leveraging wireless technologies have been explored to address the indoor localization problem [1] [4] [5]. In recent years, many researchers have studied a wide range of indoor localization systems using Wi-Fi received signal strength indicator (RSSI) fingerprints [1] [4] [5]. These systems were developed based on fingerprinting technique and WiFi technology. The main idea of fingerprint approach is to collect wireless radio features (i.e. fingerprints) at every location (i.e. reference points – RPs) in the area of interest and then build a fingerprint database. The location of an unknown user is then estimated by mapping the fingerprints collected at the user's device against the database. Because of the high availability of WiFi-enabled smartphones and portable devices and the high coverage of WiFi in indoor environments, WiFi is an ideal candidate for fingerprinting approaches [1] [2] [4]

[6]. In these systems, received signal strength indicator (RSSI), which quantifies the strength of the signal as it arrives at the user's device from the WiFi access point (AP), is widely used as the signal metric for building the fingerprint database and matching for the user's location. Machine learning methods are frequently employed to match WiFi RSSI measurements with their corresponding locations [1] [2] [4] [6].

Location	WiFi RSSI Fingerprint
(x, y)	[(AP_0, RSSI_0), (AP_1, RSSI_1),..., (AP_n, RSSI_n)]

Figure 2 An example of WiFi RSSI fingerprint

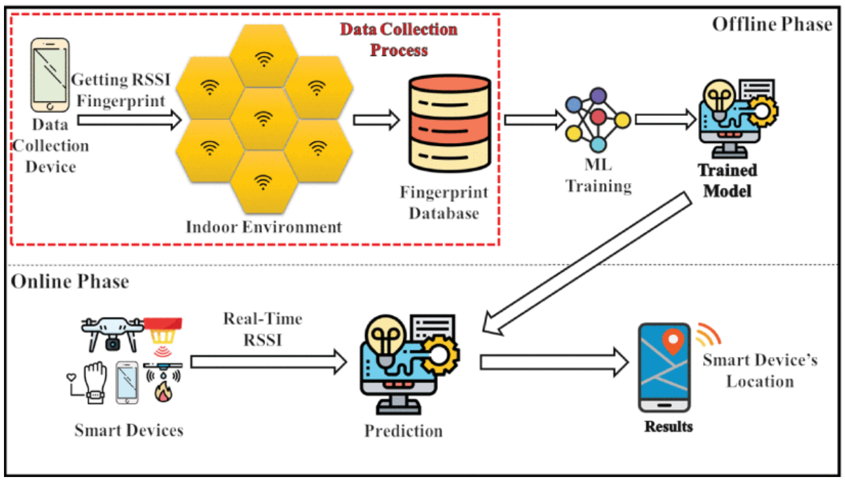

Figure 3 Basic workflow of an ML-based indoor localization system using Wi-Fi radio map [5]

Figure 2 illustrates an example of a WiFi RSSI fingerprint. In this representation, the location corresponds to the user's coordinates within the area of interest, and it is linked to a list of (AP, RSSI) pairs, each of them indicating the RSSI measurement recorded by the user's device from the respective AP. Figure 3 describes a basic workflow of a general ML-based indoor localization system using WiFi fingerprinting approach [5]. The system consists of two phases: offline phase and online phase. In the offline phase, a WiFi radio map (i.e. fingerprint database) is constructed by collecting WiFi RSSI measurements at various location points inside the indoor environment of interest. The database is often used for training a ML-based localization model. In the online phase, the model is used to serve localization queries from users. When a user sends a location query with WiFi RSSIs measured from smartphones or portable devices associated with him, the WiFi RSSIs are matched to the WiFi radio map to find the best possible location of the user. The matching process is typically performed by the trained ML model.

One of the key challenges of WiFi fingerprinting based approaches is **the construction of radio map**, which is time-consuming, labor-intensive and vulnerable to environment dynamics [7] [8] [9]. To achieve high positional accuracy, the radio map should have a high coverage level. In terms of spatial coverage, the number of reference points in the WiFi radio map should be large enough. In terms of temporal coverage, at the same RP, WiFi fingerprints should be collected at different points in time to capture changes or variations over time. In terms of device diversity, because different mobile devices can report different RSS readings even at the same RP [10] [11] [12] [13], the WiFi RSSIs should be collected from a variety of devices for better system generalization. Further, the WiFi radio map needs to be repeated for each new area and also every time there is a significant change in the area (e.g., when new APs are added or existing ones relocated) [7] [8] [9]. In a traditional data collection process, surveyors have to collect and annotate WiFi RSSI measurements at one RP for some time to reduce variation [14] [8] [7]. Taking various requirements of radio map coverage into account, the construction of a WiFi radio map for indoor localization necessitates a substantial amount of effort.

Beside simulation-based and active crowdsourcing, passive crowdsourcing has been proposed to alleviate the burden of the data collection workload and shown potential for gathering WiFi fingerprints to address real-world indoor localization challenges. However, to realize this capability, this method needs to prove its effectiveness beyond the laboratory setting. Typically, to evaluate the WiFi radio map collected through passive crowdsourcing, researchers set up an application for automatic data collection and simultaneously manually collect and label a reference dataset for evaluation purposes [13] [14]. The WiFi radio map constructed from passive crowdsourcing methods are then compared to the manually collected dataset. Due to resource limitations, this approach often results in evaluation outcomes restricted to small-scale experimental environments. To demonstrate a method's capabilities, evaluating its effectiveness on a larger scale setting is crucial. Based on the reviewed literature [13] [15] [16] [17] [8] [18] [7] [9] [10], no studies have assessed the performance of passive crowdsourcing methods on a large scale setting. With the recent availability of large-scale indoor localization dataset (e.g. the Microsoft Location Competition 2.0 Dataset [19] comprises 100,000 RPs, with WiFi RSSIs, and inertial sensor data collected from 30,000 trajectories within over 200 buildings), evaluating the large-scale empirical performance of passive crowdsourcing methods for WiFi radio map construction is feasible. Therefore, the primary task of this study is to **evaluate the performance of passive crowdsourcing methods for WiFi radio map construction on large-scale settings**. Further, a key element in passive crowdsourcing methods involves using motion models to estimate motions of users, including step counting, heading estimation and stride length estimation. These models, based on data from inertial sensors, tend to exhibit increasing distance estimation errors over time and distance traveled [17] [20] [21]. This error directly affects location accuracy and, consequently, the quality of the constructed WiFi radio map. In smaller spaces like single office floors, the error remains minimal due to shorter distances and closer reference points (RPs). However, in larger areas such as shopping malls and longer journeys, the error can become significant, impacting data quality. Thus, the second task of this study is to **investigate and assess the error in motion models used by passive crowdsourcing methods on large-scale settings**.

Generally, the main contributions of my research are as follows:

- I test and evaluate passive crowdsourcing methods for WiFi radio map construction in large-scale environments. Specifically, I implement, test and evaluate a passive crowdsourcing method, namely **Zee** from the highly influential paper [7] on a large-scale indoor localization dataset for malls, namely **Microsoft Indoor Location 2.0** dataset [19]. Based on the reviewed literature, this is the first research attempt to conduct experiments in large-scale malls.

- I implement and evaluate multiple motion models commonly used in passive crowdsourcing. These models are tested on various tracks collected from megamalls.

- I open-source my implementation[1] for reproducibility and accelerating innovation purposes. Researchers can reproduce and adapt my code for their specific needs, accelerating their work on this topic.

1.2 Research Motivation

Table 1 reviews some essential aspects of key passive crowdsourcing methods for WiFi radio map construction. In this table, the data column indicates the data directly used for localization of WiFi fingerprints, the testing area column presents basic characteristics of the testing environment, the motion model column indicates whether the method uses a motion model for estimate the user’s walk and the floor map column marks whether the method uses the floor map for WiFi radio map construction.

[1] https://github.com/hungnd11/my-thesis/

System	Data	Testing area	Motion model	Floor Map
Zee [7]	IMU sensors	Office, 65m x 35m	Yes	Yes
UnLoc [18]	IMU sensors	Office, 1750m^2; Office, 3000m^2; Mall, 4000m^2	Yes	No
LiFS [8]	IMU sensors, WiFi RSSI	Office, 70mx23m	Yes	Yes
EZ [10]	WiFi RSSI	Office, 27mx18m; Call center, 140mx90m	No	No
WicLoc [9]	IMU sensors, WiFi RSSI	Office, 40mx30m	Yes	Yes
PiLoc [17]	IMU sensors, WiFi RSSI	Office, 75mx40m; Research Lab, 10mx12m; Library, 44mx24m	Yes	No
[16]	IMU sensors	Office, small[2]	Yes	No
WILL [15]	IMU sensors, WiFi RSSI	Office, 70mx23m	Yes	Yes

Table 1 Passive crowdsourcing for WiFi radio map construction

As can be seen from Table 1, all of the passive crowdsourcing methods (except EZ) uses IMU sensor readings in combination with motion models. These methods are frequently tested on small office floors, which is often limited to under 10000m^2. These two insights are the key to motivations of this research:

- Firstly, as passive crowdsourcing is shown to be a class of promising methods for WiFi radio map construction [7] [8] [9] [10] [17] [18] [16] and most of real-world indoor localization applications are deployed in large areas such as megamalls [22] [3], to bridge the gap between laboratory and real-world application, it should be tested on large-scale environments.

- Secondly, motion models are an essential part of passive crowdsourcing methods based on IMU sensor readings. A motion model detects steps and estimates the stride length and direction of movement in each step of a pedestrian. However, motion models suffer from an increase in accumulated errors when the pedestrian moves for a long period of time. The error may reduce the effectiveness of passive crowdsourcing methods. For this reason, performance of motion models should also be evaluated on large-scale environments.

[2] Indicated by the authors of the paper

- Thirdly, with the recent availability of massive indoor location dataset, i.e. Microsoft Indoor Location 2.0 in 2021 [19], it is possible to test passive crowdsourcing methods on large-scale environments.

In general, the motivation of this research is to bridge the gap between laboratory and real-world deployment of passive crowdsourcing for WiFi radio map construction by evaluating passive crowdsourcing methods in large-scale environments, which is possible due to the publicity of a recent massive indoor location dataset.

1.3 Research Objectives

In order to assess the gap in real-world application of passive crowdsourcing for WiFi radio map construction, the first objective of this research is to evaluate the performance of passive crowdsourcing in large-scale environments. Specifically, this research studies the performance of passive crowdsourcing on malls, where solutions for indoor localization are being of interest by industries for real-world applications [22] [3].

The second objective of this study is to evaluate the performance of motion models in IMU-based passive crowdsourcing methods. Motion models are an essential part of IMU-based passive crowdsourcing [7] [8] [17] [23] [18] [15] [16], which is responsible for inferring the relative path of the user. Specifically, this research aims at studying the error in traveled distance of these models while the user is moving around the indoor environment.

1.4 Thesis Organization

My thesis is organized into five main chapters as follows:

Chapter 1 presents an essential background to my research, my research motivation and the objectives of my research. It ends with the organization of the thesis.

Chapter 2 describes the literature review. In related work, I summarize the existing work related to evaluation of passive crowdsourcing for WiFi radio map construction. The background section provides the fundamental knowledge to implementation and evaluation of these methods.

Chapter 3 presents the methodology of this research. In this chapter, I present the workflow for evaluation, which include evaluation objectives, evaluation scenarios, evaluation metrics and requirements about the dataset for evaluation.

Chapter 4 is dedicated for experimental study. This chapter starts by describing experiment setup and then presents experiment results and analysis.

Chapter 5 summarizes contributions of the research and presents directions for future work.

Chapter 2: Literature Review

2.1 Related Work

2.1.1 Indoor Localization

Indoor localization is the process of obtaining a device or user location in an indoor environment. Over the past few decades, numerous localization techniques leveraging wireless technologies have been explored to address the indoor localization problem [1] [4] [5]. Fingerprinting techniques stand out as the most popular localization methods, while WiFi, ZigBee, and Bluetooth are the preferred wireless technologies for indoor positioning [1] [4] [5].

2.1.2 Fingerprinting-based Approaches to Indoor Localization

In recent years, many researchers have studied a wide range of indoor localization systems using fingerprinting techniques [1] [4] [5]. The main idea of fingerprint approach is to collect wireless radio features (i.e. fingerprints) at every location (i.e. reference points – RPs) in the area of interest and then build a fingerprint database. The location of an unknown user is then estimated by mapping the fingerprints collected at the user's device against the database. Because of the high availability of WiFi-enabled smartphones and portable devices and the high coverage of WiFi in indoor environments, WiFi is an ideal technology for fingerprinting approaches [1] [2] [4] [6].

WiFi fingerprint-based localization approaches depend on a collected WiFi radio map, i.e. the training dataset mapping from WiFi fingerprints to locations. The construction of WiFi radio map tends to be labor-intensive and time-consuming, as it requires manual annotation of WiFi fingerprints and it has to be repeated for every new space and each time there is a significant change to a given space (e.g., a change in AP placement) [7] [1] [2] [4] [6].

2.1.3 Approaches to WiFi Radio Map Construction

Various methods were proposed to alleviate the burden of constructing WiFi radio maps for indoor localization. These methods can be divided into three broad categories, namely simulation-based, active crowdsourcing and passive crowdsourcing:

- **Simulation-based** methods aim at predicting the WiFi RSSI values at unknown locations by interpolation or by indoor propagation modeling [24]

[10]. The primary challenge of this family of methods is the difference between the simulated WiFi radio map and the empirical WiFi radio map.

- In **active crowdsourcing**, individuals who have installed a smartphone application are prompted to associate WiFi RSSI measurements at their smartphones with their known locations on a floor plan, similar to placing a pin on a displayed map, as they move within the area of interest [13]. As these methods necessitate data collectors to associate a location with each measurement, there is a potential for data quality to be impacted by human errors. This can occur when a user intentionally or unintentionally assigns a location to a measurement. Additionally, participants may find it inconvenient as they are frequently prompted to respond to inquiries about their location. [13] [8] [9].

- **Passive crowdsourcing** aims at performing location annotation without any user intervention. The idea of these methods are that, with the user's permission, a background application is installed on user's smartphones with the task of autonomously collecting WiFi RSSI measurements and associating locations with these measurements [7] [8] [9] [13] [17]. These methods exploit sensor readings from inertial sensors available in smartphones. Based on the values collected from these inertial sensors over time, a user's movement model is constructed, incorporating constraints on physical location to determine the most feasible user trajectory. Consequently, the locations of WiFi RSS measurement samples can be determined. As this group of methods doesn't require active location labeling by participants, it effectively addresses the challenges encountered in active crowdsourcing methods. Additionally, the WiFi radio map is collected from real-world scenarios, which eliminates the issues encountered by simulation-based methods. The primary challenge within this group of methods lies in the accuracy of the constructed WiFi radio map when compared to reality.

2.2 Background

2.2.1 Inertial Sensing

Inertial sensing for localization involves the use of sensors, typically accelerometers, gyroscopes, magnetometers, to track the motion and orientation of a device in the space of interest. Inertial sensors, which are being equipped in almost every modern smartphone, enable measurement of direction, acceleration, rotational velocity, and altitude. If the starting location of a user is known, a device can be tracked using dead-reckoning, wherein the inertial sensor measurements are integrated over time [7] [25] [20]. Key challenges in motion estimation by using inertial sensor readings are the complexity of human locomotion [7] and the accumulated errors over time [21] [17] [20] [25].

2.2.2 Passive Crowdsourcing for WiFi Radio Map Construction

Several passive crowdsourcing works have been proposed for automatic WiFi radio map construction. These studies can be categorized into two classes based on their utilization of floor maps.

Passive Crowdsourcing with the Aid of Floor Map

Regarding the use of floor maps, Zee [7] employs physical constraints on the floor map, such as pathways, entrances, and walls, to derive the most plausible routes of the user. Two critical components in Zee are the Placement Independent Motion Estimator (PIME) and Augmented Particle Filter (APF). PIME utilizes inertial sensors to estimate user's motion, while APF combines this motion estimation with the floor map as input for user localization. By using inertial sensor data, PIME estimates motion by determining step counts (by using a placement-independent model) and direction of movement. As the user moves, APF tracks and updates the probability distribution over possible user locations on the floor map. After a sufficiently long period of time, Zee can provide a time-series estimate of the user's locations with the highest probability. This location sequence is then used to build a WiFi fingerprint database. The work of Li et. al. in [23] share a similar idea, in which inertial sensor data and floor map constraints are combined to provide the best location estimates of the user over time.

LiFS [8], WILL [15], WicLoc [9] is a class of methods which exploits the special structure mapping between physical space and fingerprint space for WiFi radio map construction. LiFS [8] divides the floor map into equally sized bins and employs the Multi-Dimensional Scaling (MDS) algorithm to transform the floor map into a stress-free representation, where distances between bins represent pedestrian walking distances in reality. In the fingerprint space, LiFS performs a similar transformation, where distances between two points in this space are estimated through step counts, which are determined by a motion model from inertial sensor readings. Subsequently, a mapping between the two spaces is executed to determine the fingerprint's location in physical space. Building on the work of LiFS, WicLoc [9] improves LiFS by incorporating an activity classification model to improve the accuracy of step counting model and employing the MDS Calibration (MDS-C) method. Evaluation results in [9] demonstrate a 20% reduction in localization error, compared to LiFS.

Passive Crowdsourcing without the Aid of Floor Map

Regarding methods that do not use floor plans, EZ [10] is based on the observation that received signal strength (RSS) patterns observed and recorded, are constrained by the physical laws governing electromagnetic wave propagation. EZ considers the positions of APs and RPs as unknowns. When a sufficient number of RSS samples are available, EZ models the location conditions within the constraints of electromagnetic wave propagation and employs a genetic algorithm to find the optimal solution. Occasionally, EZ uses reliable GPS samples (when users move towards entrances/windows) as reference points to refine the solution. Although it doesn't require floor plan information, EZ's computational demands are substantial.

Another approach is taken by PiLoc [17], which exploits motion models built from inertial sensor data to construct a floor map from user movement trajectories. PiLoc breaks a user's movement trajectory into segments based on the analysis of sensor data, such as using a turning point to divide a trajectory into two or more segments. Multiple movement segments from various users are then combined to represent the physical space of interest. A physical floor map is constructed from these segments, allowing fingerprints within trajectories to be annotated with locations. UnLoc [18], [16] are another class of approaches based on inertial sensors, which exploits dead-reckoning to learn virtual indoor landmarks that exist in the environment to aid localization. Some locations of virtual landmarks need to be known to be able to construct WiFi radio map for the whole area.

2.2.3 Particle Filter

A particle filter is a computational technique, which is primarily used for state estimation in dynamic systems, particularly in scenarios where the system's state is not known with certainty and must be estimated from noisy or incomplete sensor data. The particle filter method is employed in various applications, such as object tracking, localization, and autonomous navigation. [26]

The core idea behind a particle filter is to represent the system's state as a set of particles or data points. These particles collectively form an approximation of the probability distribution of the state. During each iteration, the particles are updated and resampled based on measurements and system dynamics. This process refines the estimate of the system's state over time, making particle filters highly adaptable to dynamic and nonlinear systems.

Chapter 3: Methodology

3.1 Problem Formulation

Passive Crowdsourcing for WiFi Radio Map Construction

A general working scenario for a smartphone-based passive crowdsourcing system for WiFi fingerprint is as follows [7] [8] [9] [17]: A pedestrian carrying a smartphone moves around an indoor environment as part of their activities. The smartphone is equipped with an application that actively scans for WiFi beacons and records this data and sensor readings from inertial measurement units (IMUs) of the smartphone with timestamps. Later, the application sends the collected data (i.e. WiFi RSS and IMU sensing data) to a centralized backend system, where a passive crowdsourcing algorithm automatically generates location-annotated WiFi measurements of the form ***(location, WiFi RSS)*** to a database. This database of measurements can then be used to localize new users using existing WiFi localization techniques [7] [8] [9] [17]. A passive crowdsourcing method is at the core of the system, which is responsible for annotating WiFi measurements with locations.

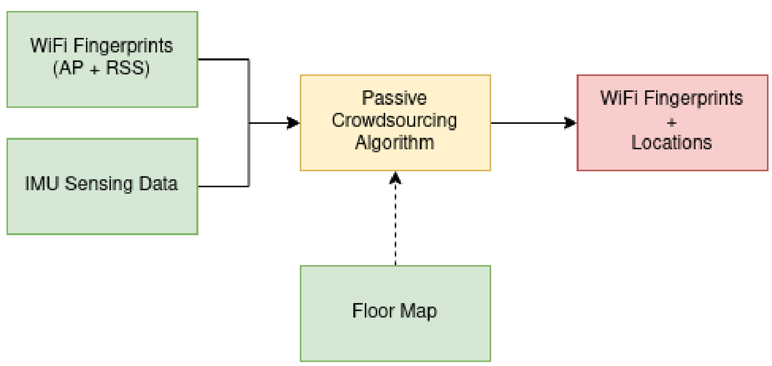

Figure 4 Input and output to a smartphone-based passive crowdsourcing method

Figure 4 visualizes inputs and outputs of a smart-phone based passive crowdsourcing method. The input consists of time-series WiFi fingerprints (i.e. WiFi access points and their associated received signal strengths measured at the smartphone side), time-series IMU sensor readings (i.e. accelerometer, magnetometer, gyroscope) and an optional floor map. The algorithm determines the locations of WiFi fingerprints and saves the results into a database, which is subsequently used for localizing new users.

Formally, the problem for a smartphone-based passive crowdsourcing algorithm can be formulated as follows: Given a collection of pedestrian tracks $T = \{T_1, T_2, \ldots, T_n\}$, each track T_i is a collection of WiFi fingerprints and IMU data $T_i = [\{t_{ij}, f_{ij}, IMU_{ij}\}]$ where j-th sample of the i-th track is recorded at timestamp t_{ij} with a WiFi fingerprint f_{ij} and the set of IMU sensor measurement IMU_{ij}, estimate the locations of collected fingerprints, i.e. estimate locations for each sample in the track $T_i = [\{t_{ij}, f_{ij}, IMU_{ij}, x'_{ij}, y'_{ij}\}]$ where (x'_{ij}, y'_{ij}) is the location of the j-th sample in the i-th track.

Motion Estimation

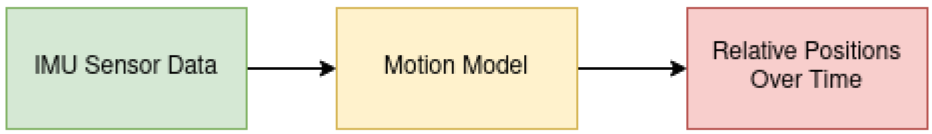

Figure 5 Input and output to a motion model

Figure 5 visualizes the inputs and outputs to a motion model. A motion model takes IMU sensor data as inputs and predicts the relative positions of the pedestrian over time. Formally, the input of a motion model is a track of IMU sensor data $T_i = [\{t_{ij}, IMU_{ij}\}]$ and the output is the relative position of the pedestrian over time $T'_i = [\{t_{ij}, IMU_{ij}, dx_{ij}, dy_{ij}\}]$ in respect to the initial position of the pedestrian.

Ground-truth locations must be collected along with WiFi fingerprints in the track to give the baseline for evaluation. For evaluation purposes, ground-truth locations are removed and the remaining data is fed into the passive crowdsourcing algorithm, as well as motion models, to get estimated locations. Then, the predicted locations are compared to ground-truth locations to evaluate the algorithm's performance. The evaluation objectives, test scenarios, and evaluation metrics will be addressed in the following subsections.

3.2 Evaluation Objectives

The first objective of this work is to evaluate the performance of passive crowdsourcing methods for WiFi radio map construction in a large-scale setting. In order to achieve the goal, it is necessary to address the performance of a passive crowdsourcing method both **individually** and **collaboratively**.

- Individually, a passive crowdsourcing method can be seen as an indoor localization algorithm and its performance is addressed by how accurate is the location annotated by the method in comparison with the manually collected data [13] [15] [16] [17] [8] [18] [7] [9] [10]. For this reason, the first evaluation objective is to find out **how well a passive crowdsourcing method is able to localize a user.**

- Collaboratively, on one hand, the performance of a passive crowdsourcing method can have direct influence upon indoor localization methods using its output [7]. On the other hand, it is influenced by prior knowledge about the environment [10] [7]. For these reasons, the second and third evaluation objectives are to find out **how well a passive crowdsourcing method performs given the prior annotated data** and **how well do existing localization schemes perform when using the location-annotated data from a passive crowdsourcing method**, respectively.

The second objective of this work is to evaluate the performance of motion models in IMU-based passive crowdsourcing methods for large-scale WiFi radio map construction. A motion model exploits data collected from an IMU integrated in the user's devices (i.e. accelerometer, gyroscope and compass) to estimate the relative positions of the user over time. In order to evaluate the performance of a motion model, we aim at finding **how well it is able to predict the relative positions of a pedestrian over time**.

3.3 Performance Metrics

This section describes performance metrics used in this study. While the localization error is used as the performance metrics for evaluating passive crowdsourcing methods, the total traveled distance error is used as the performance metrics for evaluating motion models.

Localization Error

Because a passive crowdsourcing method can be viewed as solving a localization task, the performance metrics applied for localization tasks can be used to measure the performance of the method. The main performance metric for this task is the accuracy of the position estimation, which is proposed by standard world [27] and widely used by developers and researchers [28] to test and evaluate localization systems.

The localization error is defined as the Euclidean distance between the true location and the estimated location. The formula for localization error in 2D space is defined as

$$RSE = \sqrt{(x - x')^2 + (y - y')^2}$$

where (x, y) is the coordinates of the actual position and (x', y') is the coordinates of the predicted position.

The localization error is often characterized by statistics such as the mean, the standard deviation, the percentiles, the mean squared error, the covariance matrix [27] [28]. In this study, the mean localization error is used as a measure of accuracy.

Absolute Percentage Error of Traveled Distance

There was no standard method for computing the error of motion models across developers and researchers in their studies [20]. In this study, for better comparison between methods, I use the absolute percentage error of traveled distance as the performance metric [20] [21] [25].

The absolute percentage error of traveled distance is defined as the proportion between the absolute difference of the length of the predicted path and the length of the actual path and the length of the actual path. The formula for the total traveled distance error percentage is defined as

$$APE = \frac{|L' - L|}{L}$$

where L is the length of the actual path and L' is the length of the predicted path.

3.4 Test Scenarios

Test Scenario 1: Localization accuracy of passive crowdsourcing without prior knowledge about the environment

The purpose of this scenario is to evaluate the localization accuracy of passive crowdsourcing when run in a completely new environment. For testing a passive crowdsourcing method in this scenario, we perform two steps:

- Run the passive crowdsourcing method on the input track dataset T (with ground-truth locations removed) to obtain predicted locations for fingerprints.
- Compare the predicted locations with ground-truth locations by computing the localization error.

Test Scenario 2: Localization accuracy of passive crowdsourcing with prior knowledge about the environment

The purpose of this scenario is to evaluate the localization accuracy of passive crowdsourcing when run in a known environment. This is associated with a practical scenario, when passive crowdsourcing is used to create a new version of WiFi radio map. The knowledge about the environment is presented in the form of distribution of locations. In practice, it can be inferred by predicting the locations of users, given WiFi fingerprints collected from the device and a prior WiFi radio map of the floor. In experiment, the known distribution is presented in the form of the distribution of the initial location of each user's track. For testing a passive crowdsourcing method in this scenario, we perform two steps:

- Run the passive crowdsourcing method on the input track dataset T (with ground-truth locations removed) with known distribution of locations to obtain predicted locations for fingerprints.

- Compare the predicted locations with ground-truth locations by computing the localization error.

Test Scenario 3: Localization accuracy of existing localization schemes using the data annotated by a crowdsourcing algorithm

The purpose of this scenario is to evaluate the localization accuracy of existing localization schemes using the data annotated by a crowdsourcing algorithm. This is associated with a practical scenario, when the data annotated by passive crowdsourcing is used for training WiFi fingerprinting localization models. For testing a passive crowdsourcing method in this scenario, we perform two steps:

- Split the original dataset T into two disjoint datasets: T_{train} - the track dataset for training purpose and T_{test} - the track dataset for testing purposes.

- Train the WiFi fingerprinting-based indoor localization model on the T_{train} dataset and test it on the T_{test} dataset to obtain the baseline result.

- Run the smartphone-based passive crowdsourcing method on the T_{train} (after removing ground-truth locations) to obtain passive crowdsourcing dataset T'_{train}. Train the WiFi fingerprinting-based indoor localization model on the T'_{train} dataset and test it on the T_{test} dataset to obtain model's performance when using a passive-crowdsourcing dataset.

- Compare the baseline model's performance to the model's performance when trained on passive-crowdsourcing-annotated dataset by using localization error metric.

I select k-nearest neighbor regression models [1] [29] [30] with multiple k values as WiFi fingerprinting-based localization models for this test scenario.

Test Scenario 4: Testing the performance of motion models

Because a motion model estimates the relative positions of the pedestrian, the testing of a motion model is performed at track level.

- For each track T_i, I fix the initial point of the track and run the motion model on the IMU sensor data to give relative position estimates of the pedestrian.
- Compare the estimated track with ground-truth track using the total traveled distance percentage error.
- Vary the length of the track to investigate the accumulated error of the predictions by motion model over time.

3.5 Data Requirements

For large-scale evaluation of passive crowdsourcing for WiFi radio map construction, a dataset must meet some requirements to be considered as eligible for the task.

Firstly, it should contain WiFi fingerprints and IMU sensor readings with floor map of the tested area. These are requirements from passive crowdsourcing algorithms, as presented in Table 1.

Secondly, the dataset must cover a large area, which is reflected in two criteria:

- The area of the test floor should be large enough. For this reason, only floors with areas greater than a predefined threshold (10000m^2 in practice) are considered as eligible for testing.
- The dataset should have a high level of granularity. This is indicated by the proportion of the floor covered by at least one WiFi fingerprint. I adopt a mechanism to measure the coverage of WiFi fingerprint samples. The floor map is divided into bins of 1mx1m. Each fingerprint collected is mapped to the nearest bin. The dataset coverage is defined as the number of bins covered by at least one fingerprint in the dataset. A threshold is selected to filter out floors with low-coverage.

Finally, dataset variety is crucial, and it is influenced by two primary factors: the number of tracks and the length of tracks. Short tracks, with a limited number of data points, tend to lack diversity due to their constrained movements, resulting in limited insights that can be used for location inference. On the other hand, long tracks offer a wider range of IMU sensor patterns and WiFi fingerprint patterns, making them more suitable for inferring locations. Therefore, datasets with a substantial number of long tracks are best suited for large-scale evaluations. To determine track length and assess the presence of numerous long tracks in a dataset, this study employs two specific thresholds.

Figure 6 An example of floor eligible for testing

Figure 6 illustrates an example floor eligible for testing. The area of the test floor is large enough (250mx175m), with high data coverage (i.e., more than 60% of the test hallways) and the tracks meet the specified requirements.

Chapter 4: Experiment

4.1 Experiment Setup

4.1.1 Dataset

In this study, **Microsoft Indoor Location 2.0** dataset [19] is used to evaluate passive crowdsourcing methods for collection of WiFi radio maps. It is a collection of large-scale real indoor location datasets, which were collected by professional surveyors on 204 buildings (981 floors), with over 30000 walking paths and 100000 WiFi and iBeacon samples [19]. The full dataset can be downloaded from Kaggle[3], as it was part of a virtual indoor location and navigation contest aiming at identifying the position of a smartphone in a shopping mall. The dataset consists of dense indoor fingerprints of WiFi, geomagnetic field, iBeacons etc., as well as ground truth (waypoint) locations collected from hundreds of buildings in Chinese cities. The data is organized into trace files, each of them corresponds to an indoor path walked by a site-surveyor. Trace files are grouped into floors, which in turn are grouped into sites. As presented in the competition's webinar[4], during a walk, an Android smartphone is held flat in front of the site surveyor's body, and a sensor data recording application is running on the smartphone to collect IMU (accelerometer, gyroscope, magnetometer) sensor readings, as well as WiFi fingerprints. The WiFi fingerprints and IMU sensor readings are acquired from Android's WiFi scanning API[5] and sensors API[6], respectively. In addition to raw trace files, the dataset also provides metadata about floor maps, including raster image, size and geometries in GeoJSON format for each floor.

[3] https://www.kaggle.com/c/indoor-location-navigation/data
[4] https://www.youtube.com/watch?v=xt3OzMC-XMU
[5] https://developer.android.com/reference/android/net/wifi/ScanResult.html
[6] https://developer.android.com/guide/topics/sensors

```
1574572524105	TYPE_ACCELEROMETER_UNCALIBRATED -0.6326904	0.1665802	8.124954	0.0	0.0	0.0	3
1574572524125	TYPE_ACCELEROMETER	-1.065445	0.66578674	7.383957	2
1574572524125	TYPE_MAGNETIC_FIELD	-31.11267	-6.0073853	-22.10083	3
1574572524125	TYPE_GYROSCOPE	0.46897888	-0.03993225	0.015365601	3
1574572524125	TYPE_ROTATION_VECTOR	0.042098075	0.009962577	-0.75962687	3
1574572524125	TYPE_MAGNETIC_FIELD_UNCALIBRATED	-92.10968	-23.747253	-368.3899	-60.99701	-17.739868	-346.28906	3
1574572524125	TYPE_GYROSCOPE_UNCALIBRATED	0.5832062	-0.10168457	-0.013000488	0.0018310547	0.0021514893	9.3078613E-4	3
1574572524125	TYPE_ACCELEROMETER_UNCALIBRATED -0.7081146	0.3790741	7.4497833	0.0	0.0	0.0	3
1574572524146	TYPE_ACCELEROMETER	-1.5167542	0.8842468	7.4336243	2
1574572524146	TYPE_MAGNETIC_FIELD	-32.518005	-8.102417	-23.460388	3
1574572524146	TYPE_GYROSCOPE	0.32092285	0.010147095	0.13467407	3
1574572524146	TYPE_ROTATION_VECTOR	0.047295004	0.0037991519	-0.758777	3
1574572524146	TYPE_MAGNETIC_FIELD_UNCALIBRATED	-93.515015	-25.842285	-369.74945	-60.99701	-17.739868	-346.28906	3
1574572524146	TYPE_GYROSCOPE_UNCALIBRATED	0.47080994	-0.03778076	0.016296387	0.0018310547	0.0021514893	9.3078613E-4	3
1574572524146	TYPE_ACCELEROMETER_UNCALIBRATED -1.4203796	0.8848572	7.380966	0.0	0.0	0.0	3
1574572524166	TYPE_ACCELEROMETER	-1.3336029	0.7866821	7.7783966	2
1574572524166	TYPE_MAGNETIC_FIELD	-34.628296	-6.0073853	-23.460388	3
1574572524166	TYPE_GYROSCOPE	0.19894409	0.03465271	0.22842407	3
1574572524166	TYPE_ROTATION_VECTOR	0.047295004	0.0037991519	-0.758777	3
1574572524166	TYPE_MAGNETIC_FIELD_UNCALIBRATED	-95.625305	-23.747253	-369.74945	-60.99701	-17.739868	-346.28906	3
1574572524166	TYPE_GYROSCOPE_UNCALIBRATED	0.3227539	0.012298584	0.13560486	0.0018310547	0.0021514893	9.3078613E-4	3
1574572524166	TYPE_ACCELEROMETER_UNCALIBRATED -1.4616852	0.83277893	7.6287537	0.0	0.0	0.0	3
1574572524224	TYPE_WIFI	intime_pos	06:74:9c:2e:9e:f3	-42	5825	1574572523661
1574572524224	TYPE_WIFI	intime_office	0a:74:9c:2e:9e:f3	-42	5825	1574572523662
1574572524224	TYPE_WIFI	cloud time_license_5	1e:74:9c:2e:9e:f3	-43	5825	1574572523664
1574572524224	TYPE_WIFI	intime_free	0e:74:9c:2e:9e:f3	-43	5825	1574572523662
1574572524224	TYPE_WIFI	intime_lease	12:74:9c:2e:9e:f3	-43	5825	1574572523662
1574572524224	TYPE_WIFI		16:74:9c:2e:9e:f3	-44	5825	1574572523662
1574572524224	TYPE_WIFI	intime_pos	06:74:9c:2e:9e:f2	-47	2432	1574572522602
```

Figure 7 An example trace file in Microsoft indoor location 2.0 dataset

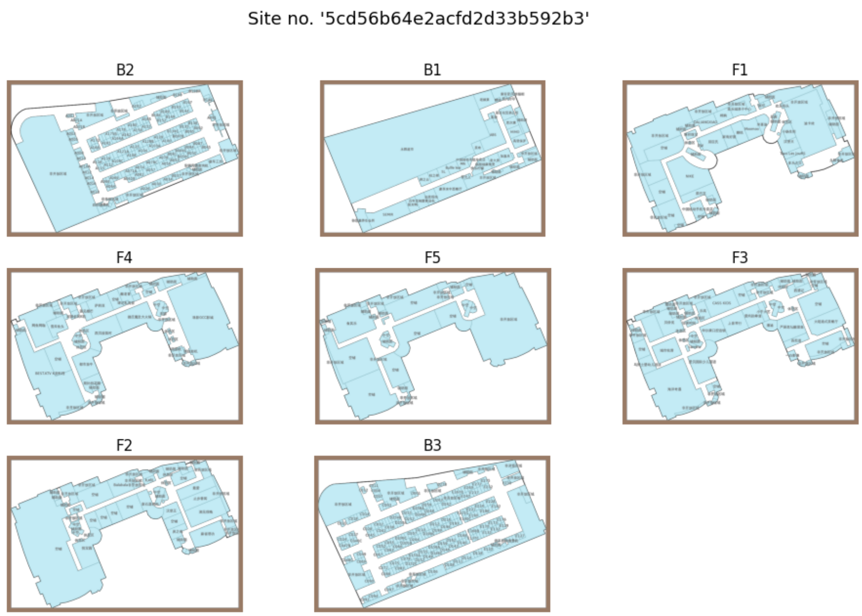

Figure 8 An example of floor maps in a site in the Microsoft indoor location 2.0 dataset

4.1.2 Zee Passive Crowdsourcing Method

In this study, I selected **Zee** from a highly influential paper [7] as the passive crowdsourcing method for evaluation. Zee exploits the IMU sensor readings and requires a floor map for WiFi radio map construction. In this section, I briefly describe the working of Zee.

Zee: Zero-effort Crowdsourcing for Indoor Localization

Zee is a system with the vision of making the WiFi radio map construction zero-effort by enabling training data to be crowdsourced without any explicit effort on the part of users. Zee leverages inertial sensors (e.g., accelerometer, compass, gyroscope) present in the mobile devices such as smartphones carried by the users, to track them as they traverse an indoor environment [7]. The key idea behind the automatic inference of location in Zee is to combine the sensor information with the constraints imposed by the map, thereby filtering out infeasible locations over time and converging on the true location.

Figure 9 depicts the overview of Zee's architecture. There are two key components in Zee: Placement Independent Motion Estimator (PIME) and Augmented Particle Filter (APF). PIME is a motion model, which uses the accelerometer, compass and gyroscope data to estimate the user's motion, which includes counting steps and estimating the heading direction of the user. A key feature of PIME is that it is independent of device placement, i.e., it is not affected by the placement of the phone carried by the user. APF is responsible for tracking the probability distribution of the user's location as he/she walks on the floor. The APF uses the steps and direction of walking from the PIME to track the user's location. To simultaneously estimate location, stride length and heading offset, the APF maintains a four-dimensional joint probability distribution function in the form of a particle filter, consisting of 2D location, stride length and heading offset, and learns all these values while the user walks on the floor. It also implements a backward-belief propagation algorithm to find the most probable trajectory of movement.

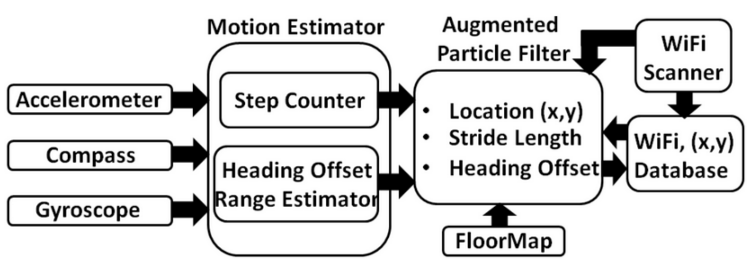

Figure 9 Zee architecture [7]

Figure 10 An example run of Zee [7]

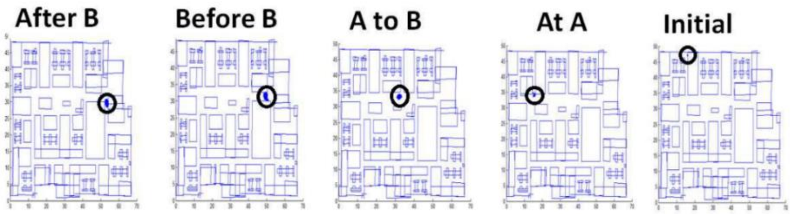

Figure 11 Backward belief propagation in Zee [7]

Figure 10 illustrates an example run of Zee in an office floor. The user walked from O to D, along the path OABCD. At first, particles were initialized at all possible locations with all possible stride lengths and heading directions. When the user walks through turns A and B respectively, as many possibilities are eliminated, the spread of particles reduces significantly and converges to a single cluster after the user takes a turn at B. As a result, the user is located. Later, a backward belief propagation algorithm, which is illustrated in Figure 11, is run to find the best matching sequence of movements. After the user's walk is localized, the WiFi radio map is constructed by inferring locations for WiFi fingerprints in each walk and combining them into a single map. In each step of a walk, the location is estimated by averaging the locations of particles in that step.

4.1.3 Motion Models

For experiments with motion models, I implement three step models and three stride-length estimation models. Because in the test dataset, the phone's location is fixed, I assume that the user's heading direction is fixed to the phone orientation. As previous work in the field [23] [17], I use orientation quaternion[7] to represent the phone orientation so that the so-called gimbal lock problem can be avoided.

Step Models

- **Local acceleration variance thresholding model** [21] [17]: This is the model implemented by PiLoc [17]. The algorithm comprises four steps. First, it computes the magnitude of the acceleration for every sample. Second, it computes the local acceleration variance to highlight the foot activity and to remove gravity. Third, two thresholds are utilized to detect swing phase with high acceleration and stance phase, respectively. Finally, a step is detected in a sample when a swing phase ends and a stance phase starts.

[7] On Android, quaternions are available through TYPE_ROTATION_VECTOR sensor readings: https://developer.android.com/reference/android/hardware/SensorEvent.html#sensor.type_rotation_vector:

- **Angular rate based model** [21] [25] [20]: The algorithm initially computes the total angular rate magnitude using gyroscope sensor readings. Then, it performs a thresholding in combination with a median filter to remove outliers. Finally, it detects transitions from a motion to motionless state.
- **Normalized Auto-correlation based Step Counting (NASC) model** [7]: This is the model implemented by Zee [7]. The intuition behind NASC is that if the user is walking, then auto-correlation will spike at the correct periodicity of the walker. For a specific lag value T, NASC computes the normalized auto-correlation at every m^{th} sample. Since the value of T is not known a priori, NASC tries to find and update optimal value T_{opt} for T during the user's walk. Then, it uses a thresholding mechanism to detect idle/walking state of the user and generates a step event every $T_{opt}/2$ samples.

Stride-Length Models

- **Random stride-length model** [7] [17]: This is the basic model used by both Zee and PiLoc. The model works by assuming that stride-length varies around an initially fixed value. So for each step, a random perturbation is added to a fixed stride length value.
- **Weinberg stride-length model** [24] [20]: The algorithm proposed by Weinberg assumes that stride length is proportional to the vertical movement of the human hip. This hip bounce is estimated from the largest acceleration differences at each step.
- **Zero Velocity Update (ZUPT) model** [21] [25] [20]: This model perform zero velocity updates every time a step is detected. At foot stance, the velocity is known to be zero, so the idea is to correct the linear velocities obtained after integrating the accelerometer.

Table 2 lists configurations for motion models in my experiments. Some parameters are from the recommended values in papers, whereas the others are selected by performing a small procedure. I adopt the idea of grid search hyperparameter tuning[8] to find the best model's configurations. Grid search is a hyperparameter optimization technique that exhaustively explores a specified range of hyperparameters for a machine learning algorithm to identify the combination that yields the best model performance. From this idea, I formulate the problem as finding a combination of model's parameters which minimizes the mean absolute percentage error of travelled distance over a set of tracks. As a motion model (without considering heading directions) consists of a step detector and a stride

[8] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

length model, I fixed the stride-length model part to find the optimal configurations for a step detector and vice versa.

Model	Parameter Name	Value
Local acceleration variance based step detection	window_size	4
	swing_threshold	2
	stance_threshold	1
NASC (ACF) step detection	t_min	40
	t_max	100
Angular rate based step detection	median_filter_window_size	8
	stance_threshold	0.4
Weinberg stride length model	window_size	16
	cutoff_frequency	3
	filter_order	4
	K	0.364
ZUPT stride length model	window_size	4
Random stride length model	initial_value	1
	noise_distribution	normal
	noise_pct	0.15

Table 2 Parameters for motion models

4.1.4 Data Preprocessing

For experimental study with Zee, the dataset is preprocessed according to requirements discussed in section 3.5. Because Zee relies on the motion of pedestrians over time and the constraints imposed by the map, only long-enough tracks are localizable by Zee. Consequently, I filter out tracks which are too short (i.e. the duration is less than 30s and the distance is less than 10m). I only select floors with at least a minimum number of tracks (i.e. 75 tracks) and a minimum specified area (i.e. 10000 m^2, which is equivalent to a 100mx100m area). For each track, the WiFi RSSI measurements and associated locations are extracted to obtain the ground-truth dataset. Because the original trace file does not give the accurate location to WiFi RSSIs, but it specifies locations of RPs (called waypoints) with timestamps, I use linear interpolation[9] to find ground-truth WiFi fingerprint locations and discard unreliable samples.

For experimental study with motion models, I select tracks which are long enough (i.e. distance is greater than 100m and duration is at least 60s). The reason behind this is that I want to investigate the trend of motion model's errors over time.

[9] https://docs.scipy.org/doc/scipy/tutorial/interpolate.html

4.1.5 Implementation and Experiment Environment

All methods were implemented in Python (version 3.10). Experiments are performed on a personal computer with system information described in Table 3.

Element	Specification
Processor	Intel Core i5 – 8250U
Memory	20 GB
Storage	512 GB of SSD
Operating System	Ubuntu 22.04

Table 3 System information of machine running experiments

4.2 Results and Analysis

4.2.1 Performance of Zee

This section presents Zee's performance by analyzing its performance on a specific floor and multiple floors. In general, experimental findings indicate that prior knowledge of the environment significantly impacts model performance and in best case, Zee's performance is acceptable for large-scale construction of WiFi radio maps.

Performance of Zee on a specific floor

Figure 12 presents the cumulative distribution function (CDF) of Zee's localization error over all WiFi fingerprints on a specific floor. The blue and orange lines present the performance of Zee with and without knowledge about distribution of initial user's locations (test scenario 1 and 2), respectively. The figure suggests that when the distribution of the initial user's location is unknown (test scenario 1), the performance of Zee is significantly worse than in the scenario when the distribution of the initial user's location is known (test scenario 2). The mean of localization error is 14m, which is acceptable for an area of 300x250m (5% of a side).

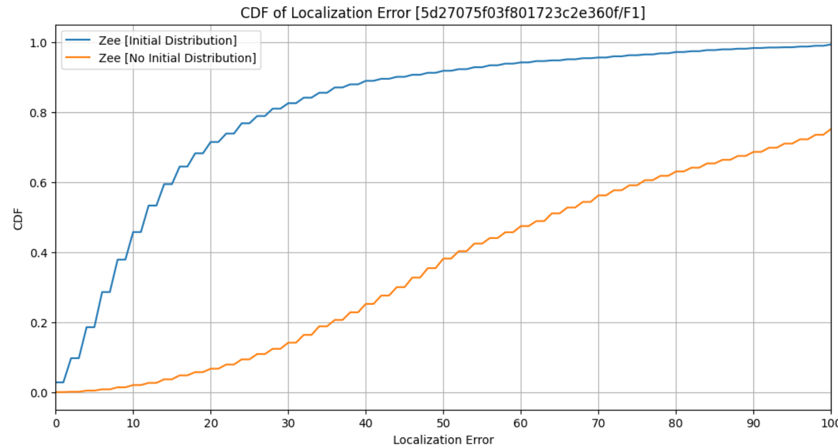

Figure 12 Zee performance on a specific floor

This phenomenon can be attributed to the fact that in a large floor, there are many possible sequences of locations which result in the same movement of users in the walk. In other words, the map filtering mechanism of Zee is not capable of finding the single most probable location from a set of possible options. This argument is demonstrated in Figure 13 and Figure 14. Without knowledge about the distribution of the initial user's location, the starting location of a walk can spread around the floor (presented as red dots in Figure 13). The user's movement, which is indicated by the red ground-truth line in Figure 16, is not enough to distinguish the actual initial locations from these options. Consequently, possible ending locations of walk after running Zee also spread around the floor in multiple clusters, which is indicated by multiple point clusters in Figure 14. When Zee is tested under the assumption that the initial location is known, the result is promising as the ending location concentrates in a single cluster (Figure 15) and the constructed walk by Zee locates near and looks similar to the ground-truth walk (Figure 16).

Figure 13 Possible initial locations of a user

Figure 14 Possible ending locations when running Zee with unknown distribution of the initial user's location

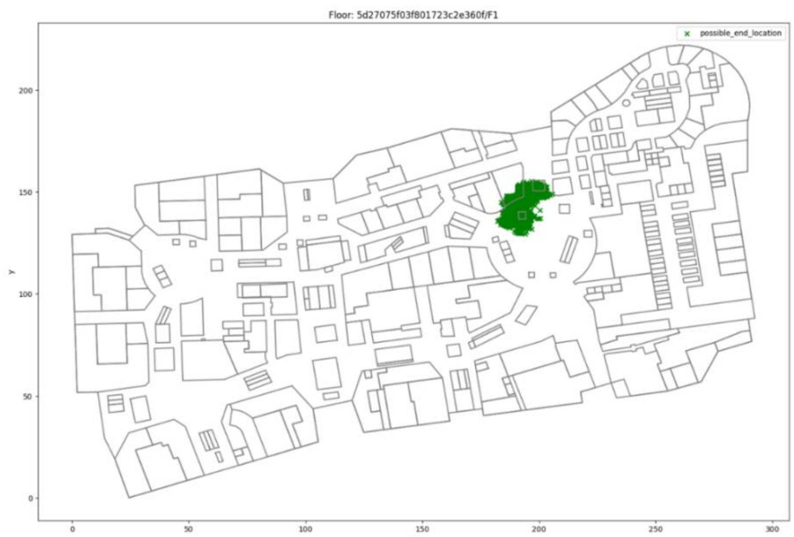

Figure 15 Possible ending locations when running Zee with known distribution of the initial user's location

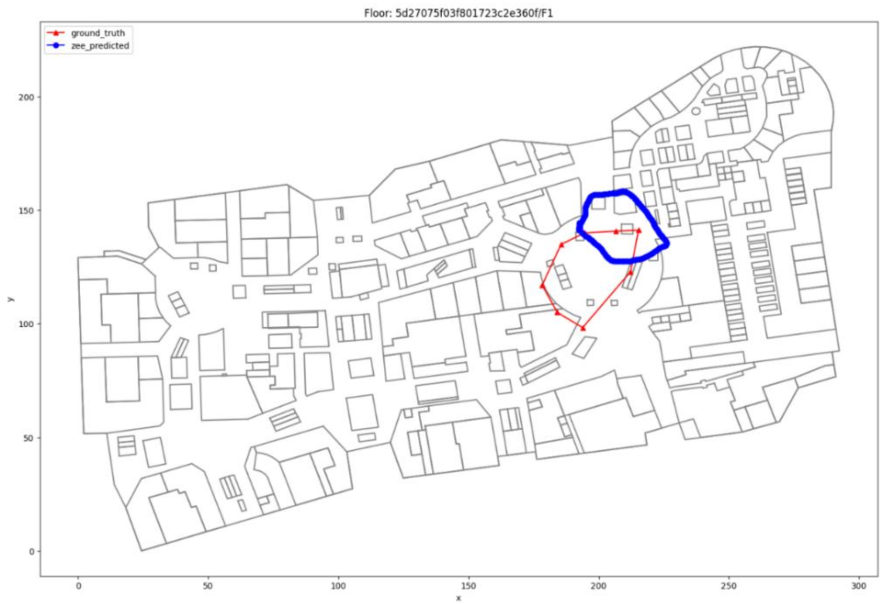

Figure 16 The constructed walk of Zee, running with known distribution of the initial user's location

In test scenario 3, I test the accuracy of k-NN[10] models when the train dataset collected by professional surveyors and when the train dataset is constructed by Zee. In both case, the best k-NN accuracy is achieved with the cosine metric and the number of neighbors equals 8. In the test with ground-truth dataset, the best mean localization error is 8m, whereas in the test with Zee-constructed dataset, the best mean localization error is 18m. Although it is 10m higher than that when trained with a ground-truth dataset, it is acceptable in an area of 300m x 250m (approximately 6% of a side).

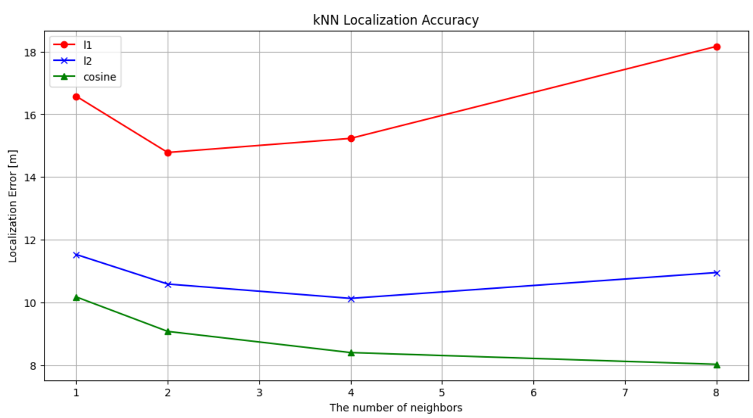

Figure 17 Mean localization errors of k-NN models when the train dataset is collected by professional surveyors

[10] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

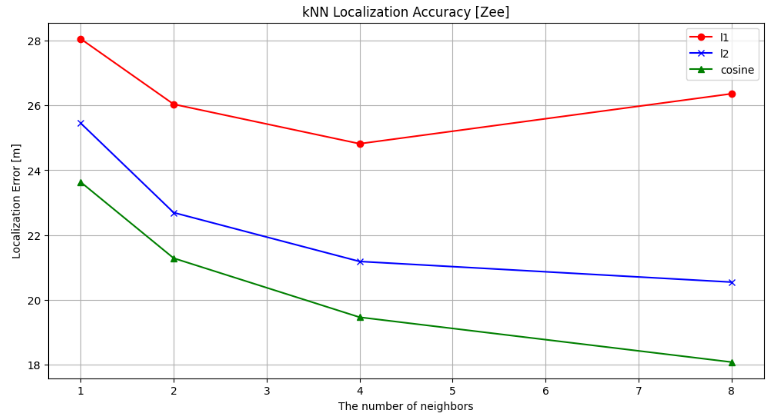

Figure 18 Mean localization errors of k-NN models when the train dataset is constructed by Zee

Performance of Zee on multiple floors

I repeat the above experiment on multiple floors and the results suggest a similar trend. Given distribution of the user's initial points, Zee's performance for WiFi radio map constructions is acceptable with a mean localization error of 18m across multiple floors. Without any prior knowledge about the environment, the performance is not better than random location predictions. When testing performance of k-NN on the WiFi radio maps collected manually and constructed by Zee, the best localization error is around 9.5m and 17.8m, respectively, which is acceptable for indoor localization in megamalls. The results of experiments on multiple floors can be seen in Figure, Figure and Figure.

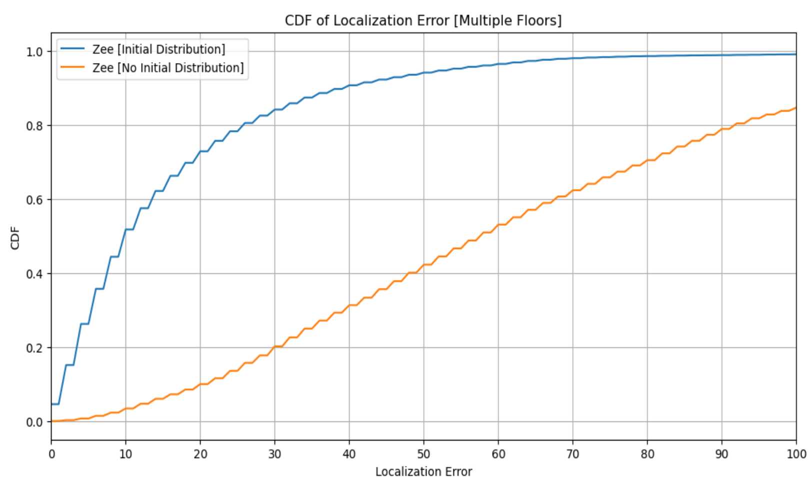

Figure 19 Zee performance, tested on multiple floors

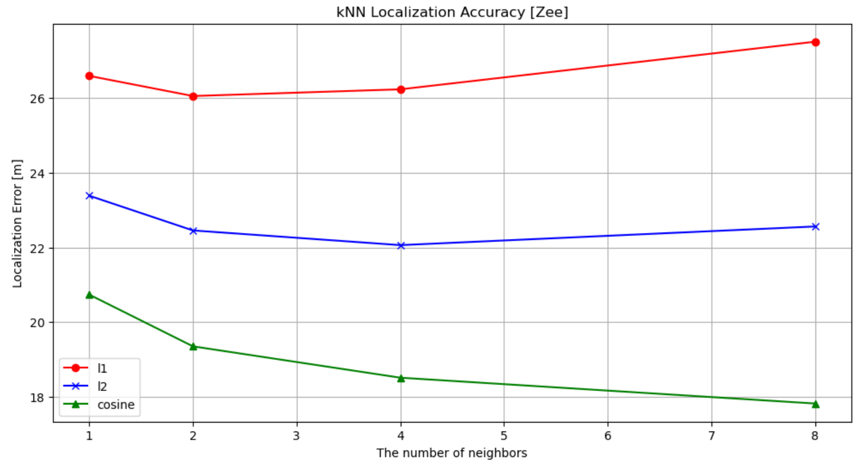

Figure 20 Mean localization errors of k-NN models when the train dataset is collected by professional surveyors (results on multiple floors)

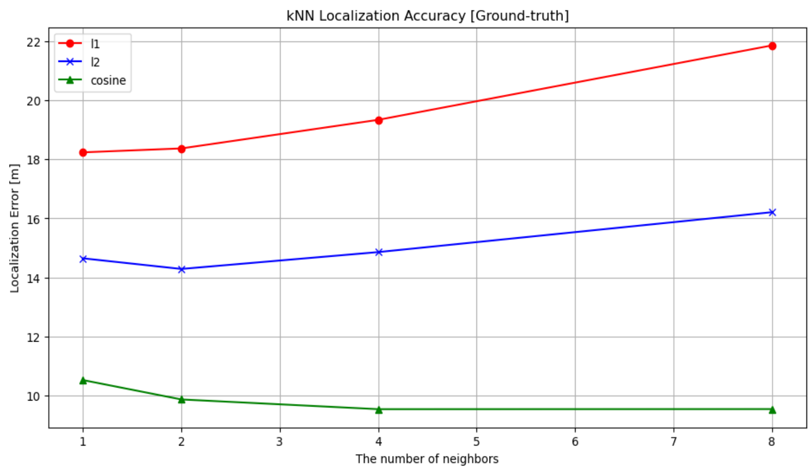

Figure 21 Mean localization errors of k-NN models when the train dataset is constructed by Zee (results on multiple floors)

4.2.3 Performance of Motion Models

Table 4 presents performances of tested motion models. The combination of NASC step detection model and Weinberg stride length model yields the minimum absolute percentage error of traveled distance at 20%. As can be seen in Figure 22, most motion models share the trend of increasing errors over time, with the exceptions of NASC + Weinberg and Angular Rate + Weinberg combinations. In general, from these observations, NASC and Weinberg are a good combination for a motion model. Figure 23 gives an illustration of results from motion models on a track, which is depicted by the blue line. The visualization shows some interesting insights, which can explain the reasons behind the performance of these models. Because ZUPT stride length model relies on the step detected from step detection model to correct the estimation, when pairing with low-accuracy step models (i.e. angular rate and local acceleration variance, which performs well when IMU sensors are installed on foot) results in inaccurate path estimation and high percentage errors. Another insight is that because NASC gives a placement-independent step model, its estimated sequence of steps are quite similar to ground-truth data, compared to other methods.

Model	Median of APE
NASC (ACF) + Weinberg	0.20
Local Acce. Var. + Random	0.26
Angular Rate + Random	0.29
Angular Rate + Weinberg	0.35
Local Acce. Var. + Weinberg	0.52
NASC (ACF) + Random	0.54
NASC (ACF) + ZUPT	0.55
Angular Rate + ZUPT	2.37
Local Acce. Var. + ZUPT	3.99

Table 4 Median of errors (APE) by motion models

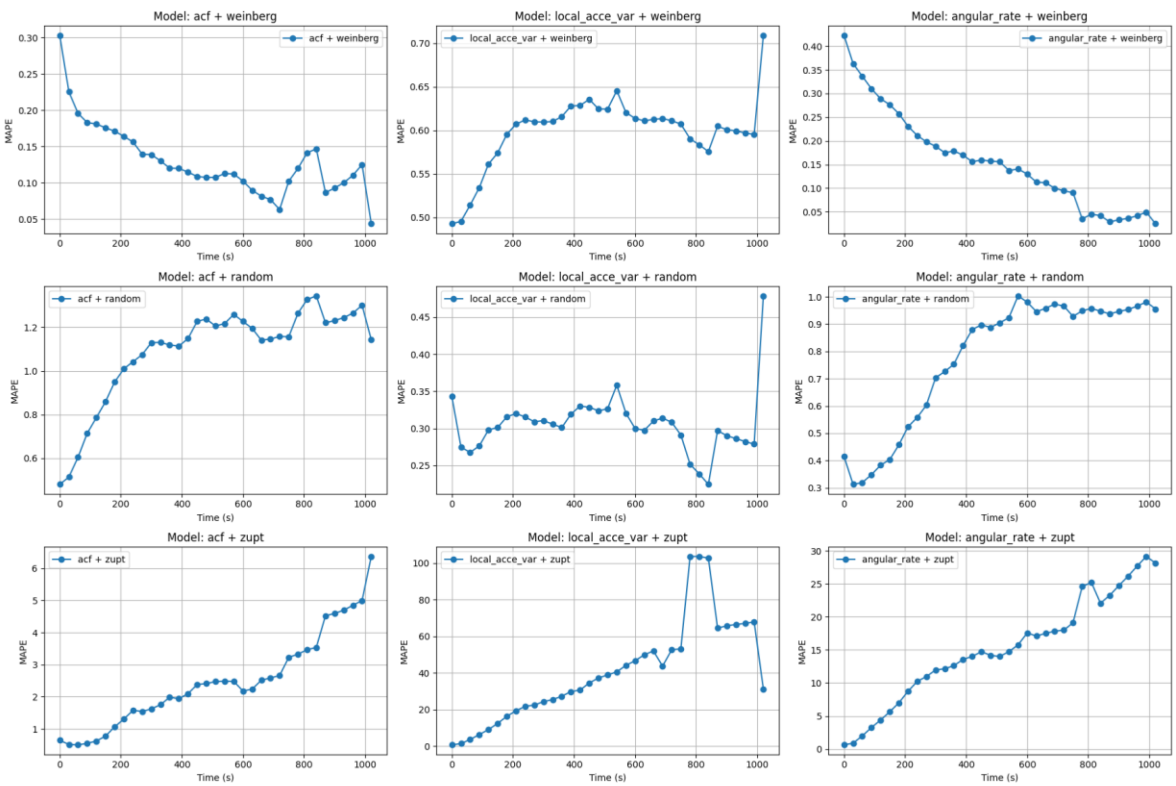

Figure 22 Performance of multiple motion models by time

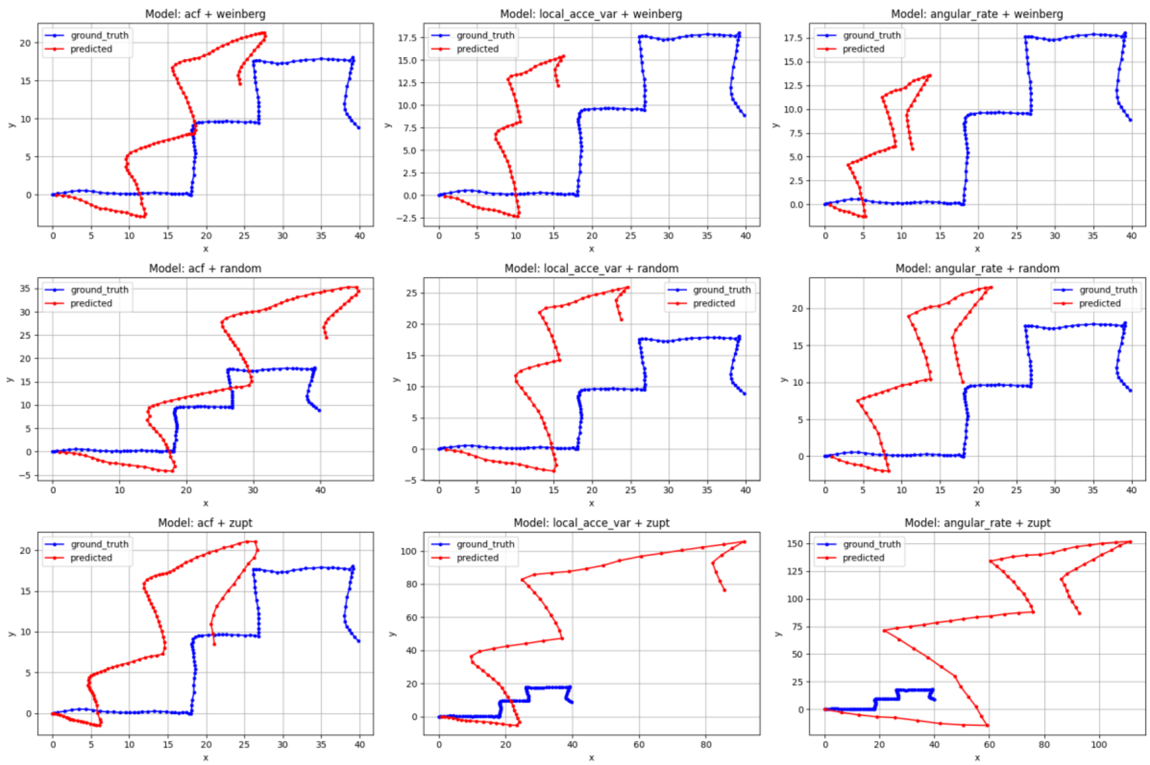

Figure 23 Results of running different motion models on a track

Chapter 5: Conclusion

5.1 Summary

In this study, I evaluate the performance of passive crowdsourcing methods for WiFi radio map construction. Specifically, I propose a workflow to test and evaluate passive crowdsourcing methods for WiFi radio map construction in large-scale environments. I implement, test and evaluate a passive crowdsourcing method, namely **Zee** from the highly influential paper [7], and multiple motions models commonly used in research on the topic on a large-scale indoor localization dataset, namely **Microsoft Indoor Location 2.0** dataset [19]..

The experiment results suggest that with prior knowledge about the area of interest, **Zee** is suitable for automatic construction of WiFi radio maps in large-scale environments. In real-world applications, **Zee** can be beneficial for collecting and updating the WiFi radio map after some stage of data initialization (i.e. collecting a WiFi radio map by professional surveyors in prior), which helps reduce the amount of work spent on data collection tasks. Moreover, the experiment results suggest that using a motion model which is placement independent helps reduce the model's tracking errors.

Finally, based on the reviewed literature, my work is the first research attempt to conduct experiments about passive crowdsourcing methods at large-scale malls. I also open-source my implementation of Zee and motion models, so that researchers can reproduce my work and adapt for their specific needs, accelerating their work on this topic.

5.2 Future Works

Regarding the future work of this study, I suggest some directions of research as follows:

- This study focuses on evaluating the performance of passive crowdsourcing methods in building a WiFi radio map on large-scale environments. This research accomplishes the target by evaluating Zee passive crowdsourcing [7] on Microsoft Indoor Location 2.0 dataset [19], which was collected in large shopping malls. A research direction is to evaluate the results on different indoor environments. This research direction can be achieved with the support from large-scale datasets.

- Recently, deep learning techniques have been widely researched and applied in practice, because of their abilities to build high-precision prediction models based on real-world data. Deep learning is also proposed for building motion models, as seen in [31]. A future research direction of this work is to conduct a comprehensive evaluation of the performance of deep learning-based motion models. Another potential direction for future work is to integrate deep learning-based motion models into passive crowdsourcing methods for WiFi radio map construction tasks.

References

[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] K. A. Nguyen, Z. Luo, G. Li and C. Watkins, "A review of smartphones-based indoor positioning: Challenges and applications," *IET Cyber-Systems and Robotics*, vol. 3, no. 1, pp. 1-30, 2021.

[3] Y. Hu, F. Qian, Z. Yin, Z. Li, Z. Ji, Y. Han, Q. Xu and W. Jiang, "Localize Yourself in Malls: An Anatomy of a Commercial Localization System with One-million Users," *GetMobile: Mobile Computing and Communications*, vol. 27, no. 1, pp. 34-38.

[4] A. Nessa, F. Hussain and X. N. Fernando, "A Survey of Machine Learning for Indoor Positioning," *IEEE Access*, vol. 8, pp. 214945-214965, 2020.

[5] N. Singh, S. Choe and R. Punmiya, "Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview," *IEEE Access*, vol. 9, pp. 127150-127174, 2021.

[6] P. Roy and C. Chowdhury, "A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, 2021.

[7] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *The 18th annual international conference on Mobile computing and networking*, New York, NY, USA, 2012.

[8] Z. Yang, C. Wu and Y. Liu, "Locating in fingerprint space," in *The 18th annual international conference on Mobile computing and networking*, New York, NY, USA, 2012.

[9] J. Niu, B. Wang, L. Cheng and J. J. P. C. Rodrigues, "WicLoc: An indoor localization system based on WiFi fingerprints and crowdsourcing," in *2015 IEEE International Conference on Communications (ICC)*.

[10] K. Chintalapudi, A. Padmanabha Iyer and V. N. Padmanabhan, "Indoor

localization without the pain," in *The 16th annual international conference on Mobile computing and networking*, New York, NY, USA, 2010.

[11] V. Bellavista-Parent, J. Torres-Sospedra and A. Pérez-Navarro, "Comprehensive Analysis of Applied Machine Learning in Indoor Positioning Based on Wi-Fi: An Extended Systematic Review," *Sensors*, vol. 22, no. 12, p. 4622, 2022.

[12] F. Firdaus, I. Fadhillah, N. A. A. and A. A. M. A. , "A Light Solution for Device Diversity Problem in a Wireless Local Area Network Fingerprint Indoor Positioning System," *International Journal of Engineering,* vol. 36, no. 4, pp. 649-658, 2023.

[13] B. Wang, Q. Chen, L. T. Yang and H.-C. Chao, "Indoor smartphone localization via fingerprint crowdsourcing: challenges and approaches," *IEEE Wireless Communications,* vol. 23, no. 3, pp. 82-89, 2016.

[14] J. Torres-Sospedra, R. Montoliu, A. Martinez-Uso, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014.

[15] Chenshu Wu, Zheng Yang, Y. Liu and Wei Xi, "WILL: Wireless indoor localization without site survey," in *IEEE INFOCOM*, 2012.

[16] F. Gu, S. Valaee, K. Khoshelham, J. Shang and R. Zhang, "Landmark Graph-Based Indoor Localization," *IEEE Internet of Things Journal,* vol. 7, no. 9, pp. 8343-8355, 2020.

[17] C. Luo, H. Hong and M. C. Chan, "PiLoc: A self-calibrating participatory indoor localization system," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 2014.

[18] H. Wang, S. Sen, A. Elgohary, M. Youssef and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *the 10th international conference on Mobile systems, applications, and services*, New York, NY, USA, 2012.

[19] Y. Shu, Q. Xu, J. Liu, R. R. Choudhury, N. Trigoni and V. Bahl, "Indoor Location Competition 2.0 Dataset," Microsoft, 12 8 2021. [Online]. Available:

https://www.microsoft.com/en-us/research/publication/indoor-location-competition-2-0-dataset/. [Accessed 01 10 2023].

[20] X. Hou and J. Bergmann, "Pedestrian Dead Reckoning With Wearable Sensors: A Systematic Review," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 143-152, 2021.

[21] A. Jimenez, F. Seco, C. Prieto and J. Guevara, "A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU," in *IEEE International Symposium on Intelligent Signal Processing*, 2009.

[22] J. Ni, F. Zhang, J. Xiong, Q. Huang, Z. Chang, J. Ma, B. Xie, P. Wang, G. Bian, X. Li and C. Liu, "Experience: pushing indoor localization from laboratory to the wild," in *the 28th Annual International Conference on Mobile Computing And Networking*, New York, NY, USA, 2022.

[23] F. L. a. C. Zhao, G. Ding, J. Gong, C. Liu and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *the 2012 ACM Conference on Ubiquitous Computing*, 2012.

[24] G. Caso, L. De Nardis, F. Lemic, V. Handziski, A. Wolisz and M.-G. D. Benedetto, "ViFi: Virtual Fingerprinting WiFi-Based Indoor Positioning via Multi-Wall Multi-Floor Propagation Model," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1478-1491, 2020.

[25] R. Feliz, E. Zalama and J. Gómez-García-Bermejo, "Pedestrian Tracking Using Inertial Sensors," *Journal of Physical Agents*, vol. 3, 2009.

[26] E. J, T. E and v. d. M. R., "Particle Filters: A Hands-On Tutorial," *Sensors (Basel)*, vol. 21, no. 2, p. 438, 2021.

[27] I. 18305:2016, "ISO," [Online]. Available: https://www.iso.org/standard/62090.html. [Accessed 1 10 2023].

[28] D. Bousdar Ahmed, L. E. Diez, E. M. Diaz and J. J. Garcia Dominguez, "A Survey on Test and Evaluation Methodologies of Pedestrian Localization Systems," *IEEE Sensors Journal*, vol. 20, no. 1, pp. 479-491, 2020.

[29] S. Saha, K. Chaudhuri, D. Sanghi and P. Bhagwat, "Location determination of a mobile device using IEEE 802.11b access point signals," in *IEEE Wireless*

Communications and Networking, 2003. WCNC 2003, 2003.

[30] Y. Hu, X. Fan, Z. Yin, F. Qian, Z. Ji, Y. Shu, Y. Han, Q. Xu, J. Liu and P. Bahl, "The Wisdom of 1,170 Teams: Lessons and Experiences from a Large Indoor Localization Competition," in *the 29th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2023.

[31] O. Asraf, F. Shama and I. Klein, "PDRNet: A Deep-Learning Pedestrian Dead Reckoning Framework," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 4932-4939, 2022.

[32] M. Youssef, A. Agrawala and A. Udaya Shankar, "WLAN location determination via clustering and probability distributions," in *The First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, 2003.

[33] F. Potorti, A. Crivello and F. Palumbo, "Evaluation of Indoor Localisation Systems: Comments on the ISO/IEC 18305 Standard," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018.

[34] Y. Hu, F. Qian, Z. Yin, Z. Li, Z. Ji, Y. Han, Q. Xu and W. Jiang, "Experience: practical indoor localization for malls," in *the 28th Annual International Conference on Mobile Computing And Networking*, New York, NY, USA, 2022.

[35] M. Zhang, Z. Fan, R. Shibasaki and X. Song, "Domain Adversarial Graph Convolutional Network Based on RSSI and Crowdsensing for Indoor Localization," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13662-13672, 2023.