WiFi Fingerprinting Based Indoor Localization with Autonomous Survey and Machine Learning

by

Minh Tu Hoang

B. Eng., Hanoi University of Technology, 2013

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

WiFi Fingerprinting Based Indoor Localization with Autonomous Survey and Machine Learning

by

Minh Tu Hoang

B. Eng., Hanoi University of Technology, 2013

Supervisory Committee

Dr. Xiaodai Dong, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Hong-Chuan Yang, Department Member

(Department of Electrical and Computer Engineering)

Dr. Daniela Constantinescu, Outside Member

(Department of Mechanical Engineering)

ABSTRACT

The demand for accurate localization under indoor environments has increased dramatically in recent years. To be cost-effective, most of the localization solutions are based on the WiFi signals, utilizing the pervasive deployment of WiFi infrastructure and availability of the WiFi enabled mobile devices. However, one of the major challenges of indoor localization is that many obstacles such as walls, furniture and moving human beings, form fluctuations of WiFi signals known as multipath interferences. Such fluctuations cause significant degradation in the accuracy of indoor positioning, which has yet to be fully overcome. In this thesis, we develop completed indoor localization solutions based on WiFi fingerprinting and machine learning approaches with two types of WiFi fingerprints including received signal strength indicator (RSSI) and channel state information (CSI).

Starting from the low complexity algorithm, we propose a soft range limited K nearest neighbours (SRL-KNN) to address spatial ambiguity and the fluctuation of WiFi signals. SRL-KNN exploits RSSI and scales the fingerprint distance by a range factor related to the physical distance between the users previous position and the reference location in the database. Although utilizing the prior locations, SRL-KNN does not require knowledge of the exact moving speed and direction of the user. Moreover, to take into account of the temporal fluctuations of RSSI, RSSI histogram is incorporated into the distance calculation. Besides, the idea of the soft range limiting factor can be applied to all of the existed probabilistic methods, i.e., parametric and nonparametric methods, to improve their performances. A semi-sequential short term memory step is proposed to add to the existed probabilistic methods to reduce their spatial ambiguity of fingerprints and boost significantly their localization accuracy.

In the following research phase, instead of locating user's position one at a time as in the cases of conventional algorithms, our recurrent neuron networks (RNNs)

solution aims at trajectory positioning and takes into account of the relation among RSSI measurements in a trajectory. Furthermore, a weighted average filter is proposed for both input RSSI data and sequential output locations to enhance the accuracy among the temporal fluctuations of RSSI. The results using different types of RNN including vanilla RNN, long short-term memory (LSTM), gated recurrent unit (GRU) and bidirectional LSTM (BiLSTM) are presented.

Next, the problem of localization using only one single router is analysed. CSI information will be adopted along with RSSI to enhance the localization accuracy. Each of the reference point (RP) is presented by a group of CSI measurements from several WiFi subcarriers which we call CSI images. The combination of convolutional neural network (CNN) and LSTM model is proposed. CNN extracts the useful information from several CSI values (CSI images), and then LSTM will exploit this information in sequential timesteps to determine the user's location. All of our proposed algorithms are demonstrated by extensive on-site experiments and are compared with several existing deterministic and probabilistic methods in literature under the same test environment.

Finally, a fully practical passive indoor localization is proposed. Most of the conventional methods rely on the collected WiFi signal on the mobile devices (active information), which requires a dedicated software to be installed. Different from them, we leverage the received data of the routers (passive information) to locate the position of the user. The problem of data insufficiency in passive indoor localization is mitigated by request to send (RTS) and clear to send (CTS) process. Furthermore, the completed localization solutions for two most popular mobile device usage scenarios, i,e., idle and transmission modes, are analyzed in details. The localization accuracy is investigated through experiments with several phones, e.g., Nexus 5, Samsung, Iphone and HTC, in hundreds of testing locations. The experimental results demonstrate that

our proposed localization scheme achieves an average localization error of around 1.5 m when the phone is in idle mode, and approximately 1 m when it actively transmits data.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

my supervisor, Dr. Xiaodai Dong, for her trust, supporting, mentoring, encouragement, and patience. She is like a lighthouse amidst the chaos, guiding me to roam in the sea of knowledge and bring me a brighter future. I feel lucky and grateful to be her student.

Dr. Tao Lu for the helpful advice and enthusiastic supports you gave me during 4 years of my PhD.

Dr. Hongchuan Yang, and Dr. Daniela Constantinescu, for spending precious time to serve as my supervisory committee. Starting from the candidate exam, I have been following their valuable suggestions and finally reach the most important milestone of my life.

all of my friends (Tina Nguyen, Chris Ng, Lai Le, Nancy Ngo, Linh Nguyen, etc.), for providing me with unfailing support and continuous encouragement throughout my years. You are my sisters and my brothers giving me love and making the most memorable 4 years in my life.

my family (my parents, uncle Tin Nguyen, aunt Hoa Truong) for continuous and unparalleled love, help and support. You encourage me to explore new directions in life and seek my own destiny. This journey would not have been possible if not for them, and I dedicate this milestone to them.

Minh Tu Hoang, Victoria, BC, Canada

DEDICATION

To those who are working on indoor localization and machine learning.

Chapter 1

Introduction

1.1 Motivations

Indoor localization has attracted much attention in recent years due to its commercial values, with market value predicted to worth 10 billion dollars by 2020 [1]. There are a large variety of the applications such as guidance, rescue operation, virtual reality game, etc. [2, 3]. For example, indoor positioning can help to guide customers in a shopping mall towards store, food court, etc., or passengers in an airport to the right terminal. In a museum, accurate indoor localization can transform a customer's phone into a virtual guide to give them contextual information based on his/her location. As GPS signal cannot penetrate well in indoor environments, various other signals have been investigated for localization purpose. Among all the available solutions, one of the promising candidates is WiFi positioning since the wide availability of WiFi devices eliminates the requirement for additional infrastructure and hardware. In general, WiFi indoor localization methods can be grouped in two categories: one is signal propagation model based ranging, which utilizes received signal strength (RSS), the time of flight (TOF) and/or angle of arrival (AOA) [4] to estimate the lo-

cation of the target; the other is fingerprinting based [1], which discriminates between locations by associating physically measurable properties as fingerprints or signatures for each discrete point. Due to the strong multipath effects, exact propagation model is difficult to obtain. Therefore, the fingerprinting approach is more favourable for the WiFi based localization.

In WiFi fingerprint, received signal strength indicator (RSSI) is widely used as a feature in localization because RSSI can be obtained easily from most WiFi receivers such as mobile phones, tablets, laptops, etc. [5]. However, RSSI has two drawbacks: instability due to fading and multipath effects and device heterogeneity, i.e., different devices have different RSSIs even at the same position [5]. In order to mitigate those problems, channel state information (CSI) is adopted to provide richer information from multiple antennas and multiple subcarriers [5, 6]. So far, CSI is only available with the specific wireless network interface cards (NIC), e.g., Intel WiFi Link 5300 MIMO NIC, Atheros AR9390 or Atheros AR9580 chipset [7]. Although having been extensively investigated in the literature, all of the localization algorithms are still facing a series of problems due to the spatial ambiguity and signal instability of both RSSI and CSI [8]. Furthermore, in order to construct a sufficient fingerprint map for accurate localization, a large number of reference points are required [9], which is time-consuming and labor-intensive [1].

In this thesis, the main application is to locate a walking human using the WiFi signals of the carried WiFi devices, e.g, smartphone, laptop, etc., with an acceptable accuracy around a few feet. We propose some advanced indoor localization solutions based on WiFi fingerprinting and machine learning approaches with two types of WiFi fingerprints including both RSSI and CSI. We also address the heavy training phase challenge with the support of an autonomous robot. The robot can navigate to a target location to collect WiFi fingerprints automatically. The following sections

are our proposed research issues.

1.2 Research Objectives and Contributions

1.2.1 Soft Range Limited K-Nearest Neighbours For Accurate RSSI Indoor Localization

In Chapter 2, our target is proposing a low complexity and practical algorithm with acceptable accuracy for indoor localization. Our soft range limited K nearest neighbours (SRL-KNN) localization fingerprinting algorithm incorporates the information of a user's previous position to conventional KNN. Since the moving speed of the user in an indoor environment is bounded, the proposed method applies a penalty function based on the physical distance between the reference point and the anchor point (user's previous position) when calculating the fingerprint distance. As a result, the spatial ambiguity problem is significantly reduced and the accuracy is enhanced. The proposed research issues and contributions are identified as follows:

1. The analysis of open challenges of conventional KNN.

2. The detailed structure of SRL-KNN to address those above challenges.

3. The on-site experimental results to prove the effectiveness of SRL-KNN and the error analysis.

4. The comparison between SRL-KNN and some of other literature methods using our dataset and a published dataset.

1.2.2 Semi Sequential Probabilistic Model For Indoor Localization Enhancement

Based on the good results from SRL-KNN algorithm in Chapter 2, we expand the idea of soft range limited factor to probabilistic method in Chapter 3. The advantages of probabilistic method are low-complexity, better accuracy than KNN and can be applied effectively in multiple kinds of fingerprints, i.e., RSSI and CSI. Since the moving speed of the user in an indoor environment is bounded, the probability of the nearer locations compared with the recent user's previous point is higher than the further ones. Therefore, in probabilistic algorithms, the Bayes' formula is modified so that it contains a memory of a recent previous location to predict the current point. The detailed research issues and contributions include:

1. The survey of the existed probabilistic methods and their disadvantages.
2. The proposed models of semi-sequential probabilistic method.
3. The on-site experimental results to prove the effectiveness of our proposed model.
4. The comparison between our proposed model and the existing methods.

1.2.3 Recurrent Neural Networks For Accurate RSSI Indoor Localization

Chapter 4 focuses on recurrent neural network (RNN) which exploits the sequential RSSI measurements and the trajectory information to determine the user's location. Although the proposed SRL-KNN algorithm in Chapter 2 is a low complexity algorithm with good accuracy, it has the constraints about the speed of the users and can not handle well the sudden changing speed scenarios. In contrasts, the proposed

RNN model does not require either the assumption of the user's bounded speed. It makes the prediction by only adjusting the internal weights through training phase. Furthermore, in order to overcome the RSSI instability, the weighted average filter is applied in both training and testing RSSI measurements. The performance is tested in different types of RNN including vanilla RNN, long short term memory (LSTM) [10], gated recurrent unit (GRU) [11], bidirectional RNN (BiRNN), bidirectional LSTM (BiLSTM) [12] and bidirectional GRU (BiGRU) [13]. The proposed research issues and contributions are identified as follows:

1. The complete structure and process of the proposed RNN solutions for indoor localization.

2. The detailed analysis of the important RNN parameters.

3. The on-site experimental results to prove the effectiveness of proposed RNN solutions and the error analysis.

4. The comparison between the proposed RNNs and other literature methods using our dataset and a published dataset.

1.2.4 A CNN-LSTM Quantifier for Single Access Point CSI Indoor Localization

Most of the conventional methods rely on RSSI to locate the user's position. Although RSSI is easy to obtain, it has the drawback of wide fluctuation and the lack of information, i.e., one router provides only one RSSI reading in a frequency at one time. Therefore, in order to locate the accurate user's position, a large number of access points (APs) are used in the experiment, i.e., 6 APs in our experimental area. Different from RSSI, CSI provides multiple amplitude and phase of WiFi subcarriers

at one time. With this rich information, a unique fingerprint can be constructed for each location to precisely locate the position in the radio map with only one AP. Convolutional neural network (CNN) is applied to pre-proceed CSI information and extract the useful information. In the later step, LSTM will use the output of CNN combined with the time step information to determine user's position. The proposed research issues and contributions of Chapter 5 are identified as follows:

1. The analysis of open challenges of using CSI for indoor localization with a single AP.

2. The detailed structures of the proposed CNN-LSTM model.

3. The on-site experimental results with 2 difference devices, i.e., laptop and smart phone, to prove the effectiveness of our proposed method.

4. The comparison between the proposed CNN-LSTM algorithms and some of other literature methods.

1.2.5 Practical Passive Indoor Localization with WiFi Fingerprints

The majority of literature work follows the active model which collects WiFi signal from mobile device to infer the user's positions [8, 14]. This active localization class requires a dedicated software installed on mobile devices to perform WiFi scanning and logging data for the localization process [15]. On the other hand, in passive localization, there is no installed software needed in user's devices to obtain fingerprint data. Instead of that, the localization process is fully implemented based on the information collected from WiFi access points (APs) [16, 17]. Chapter 6 focuses on 2 most practical passive indoor localization scenarios including idle and transmission

scheme. In idle scenario, the WiFi of the mobile device is on but the user is not using the phone at the moment (e.g., the user is moving and putting the phone in their pocket). Due to the limited sending frame in that mode, request to send (RTS) and clear to send (CTS) mechanism is utilized to frequently obtain the RSSI data for localization process. On the other hand, in transmission scenario, the mobile device is connected to an AP and sending data frames (e.g., the user is watching online video or browsing web). Both of RSSI and CSI fingerprints are available in this case. The detailed research issues and contributions include:

1. The survey of the existed active and passive localization methods and their disadvantages.

2. The proposed models of comprehensive practical passive indoor localization schemes with detailed analysis of different work modes of the phone, i.e., idle, on-off screen, data transmission mode.

3. Two most popular WiFi indoor localization scenarios, including idle and transmission schemes along with RTS/CTS process utilization, are fully analyzed.

4. The proposed models are tested in an extensive autonomous experiment with the support of the robot, which includes thousands of RPs, testing trajectories and a variety of phone types such as Samsung Galaxy S6, HTC OneX, Iphone X and LG Nexus 5.

Chapter 2

Soft Range Limited K-Nearest Neighbours For Accurate RSSI Indoor Localization

2.1 Introduction

Fingerprinting based WiFi localization can be realized by deterministic and probabilistic approaches [1]. The former uses a similarity metric to differentiate the measured signal and the fingerprint data in the database before estimating the user's position as the closest fingerprint location in the signal space. Some typical examples of this approach are artificial neural network (ANN) [18], [19], support vector machine (SVM) [18], [20] and K nearest neighbors (KNN) [21], [22], all of which require the collection of the fingerprints in the training phase to be compared with the measured signal in the testing phase for localization. Compared to SVM and ANN, KNN has the lowest complexity while its accuracy is comparable to SVM [18].

On the other hand, the probabilistic algorithms are all based on statistical in-

ference between the target signal measurement and stored fingerprints using Bayes rule [23]. Moreover, improvement of localization accuracy has been achieved by exploiting the measurements in previous time steps. For example, Kalman filter [24–27] is used to estimate the most likely current location based on prior measurements, assuming a Gaussian noise and linear motion dynamics. In real scene, however, the assumption of Gaussian noise is not necessarily true [28], neither is the user's linear motion assumption a good approximation. A better motion model was proposed in [26] with two Kalman filters, one for constant velocity case and the other one for a greater acceleration. The application of these two filters and switching in-between them increases the computational complexity significantly. In order to tackle the non-Gaussian and non-linear cases, extended Kalman filter [13] or particle filter [29–32] can be applied. However, the major drawback of those filters is associated with high computational workloads and failure due to sample impoverishment [28, 31].

This chapter focuses on the study of KNN because of its low complexity suitable for practical use. In general, KNN computes the distance between the current WiFi RSSI fingerprint and the learned fingerprint in database to determine K nearest neighbours. Different distance metrics such as Euclidean distance, Manhattan distance, and Mahalanobis distance can be used in KNN [2]. Although being extensively investigated in literature, KNN still has the following open challenges:

1 Spatial ambiguity [33]: Some physically distant locations may have similar fingerprints or similar fingerprint distances compared with the current location. This could mislead the KNN algorithms, leading to high localization errors.

2 RSSI instability: Moving objects, constantly varying electromagnetic wave landscape in ambient environments, directionality of antenna and RF interference, etc., contribute to the wide fluctuation of WiFi signal [28]. Therefore, the observed fingerprint of a location in the testing phase may not match that collected

in the training phase.

3 RSSI short collecting time per location: Usually RSSI instability can be mitigated by taking the average of a large number of RSSI readings at one location. However, due to the mobile nature of the locating target, the RSSI sampling at each specific location in the testing phase is typically shorter than 2 seconds. Within that duration, only a few number of RSSI readings can be collected. Consequently, the localization accuracy is severely impaired.

4 Heavy initial training phase: in order to construct the sufficient fingerprint map for accurate localization, a large number of reference points are required [9], which is time-consuming and labor-intensive [1].

To address the first three challenges, this chapter incorporates the information of a user's previous position to KNN. Since the moving speed of the user in an indoor environment is bounded, the proposed soft range limited K nearest neighbours (SRL-KNN) algorithm applies a penalty function based on the physical distance between the reference point and the anchor point (user's previous position) when calculating the fingerprint distance. As a result, the spatial ambiguity problem is significantly reduced. In contrast to other approaches such as Kalman filters that also exploit the measurements from previous time steps [24–27], our SRL-KNN method is much simpler and does not require the assumption of Gaussian noise distribution or linear motion. In addition, this chapter proposes to use histogram and the combination of multiple fingerprints such as the mean, the difference of RSSI, the ranks of the AP RSSIs to tackle the RSSI instability and improve the localization accuracy. Actual on-site experiments show that our proposed algorithms can work well with the limited number (1 or 2) of RSSI scans in each testing location (Section 2.3.3).

To reduce the work load in the initial training phase, crowdsourcing-based ap-

proaches have been proposed to replace the professional site survey with explicit and unprofessional user participation [13, 34]. However, these methods are vulnerable to imperfect data, since the involved users are not always accustomed to the collecting systems [1]. On the other hand, [9, 32] utilize relative RSSI differences among various access points (APs) called AP-sequence to reduce the number of required reference points (RPs). In [9], the area of interest is divided into a set of small regions based on AP-sequence and the user's location is estimated to be at the center of these regions. The main disadvantage of this approach is that the localization accuracy varies widely at different regions. Therefore, in addition to WiFi AP-sequence, [32] also adopts the inertial-measurement unit (IMU) sensors and FM signal to refine the estimated location. In our experiment, we address the training phase challenge with the support of an autonomous robot. Our 3-wheel robot (Fig. 2.2(a)) has multiple sensors including wheel odometer, an inertial measurement unit (IMU), a LIDAR, sonar sensors and a color and depth (RGB-D) camera. The robot can navigate to a target location to collect WiFi fingerprints automatically. The localization accuracy of the robot is 0.07 m $\pm$ 0.02 m. Therefore, the time consumption and degree of human involvement for fingerprinting map construction is significantly reduced.

The rest of this chapter is organized as follows. Section 2.2 introduces related works on KNN, followed by details of SRL-KNN in Section 2.3. Section 3.4 reports the experimental set-up and results for the performance evaluation. Finally, Section 3.5 concludes this chapter.

2.2 Related Work

The original research on KNN indoor localization dates back to 2000 when a group from Microsoft demonstrated RADAR [21]. In that work, the mean and standard

deviation of RSSI from multiple base stations are collected in the training phase and the Euclidean distance is used in the testing phase to determine the user's position. There are 70 reference points (RPs) with 2.8 m distance spacing (grid size). Testing points are picked randomly among these reference points. The average accuracy of this system is around 3 m with 75% of the localization errors are below 4.7 m.

A refinement of the above method is the weighted KNN (WKNN) proposed by Brunato *et al.* [18], which calculates the user's position by the weighted average of the RSSI distance between estimated nearest neighbours and the current measurement. The experiment is implemented with 207 reference locations, 50 testing locations and a grid size of 1.7 m. The accuracy of WKNN is 3.1 ± 0.1 m and 75% of the localization errors are below 3.9 m.

To accommodate device heterogeneity, Zou *et al.* [35] proposed signal tendency index - weighted KNN (STI-WKNN) by adopting the similarity index STI between RSSI curve shapes to improve the localization accuracy. The raw RSSI signal is first transformed to a normalized object based on procrustes analysis (PA) method [36]. Then signal tendency index is computed according to Euclidean distances between real time PA object and those stored in the fingerprint database. The final location will be determined by weighting among K nearest neighbours that provide the smallest STI. Their experiment shows that STI-WKNN improves the localization accuracy by 23.95% over the original WKNN across heterogeneous mobile devices.

In a following research, Shin *et al.* [37] proposed to dynamically change the number of nearest neighbours K. Firstly, the RSSI Euclidean distance D_i of each reference point i is computed and N numbers of which smaller than a threshold T are picked. In a second step, the average of the selected D_i is calculated to obtain a value E and K neighbours that satisfy $D_i < E$ are chosen. In general, this method only provides slightly lower average localization error than the classical KNN in RADAR [21], except

in the corridor where the testing scene is de facto one-dimensional.

Taking into account the limited movement capabilities of a mobile user in an indoor environment, some researchers tried to utilize the information from the previous locations to improve the accuracy of KNN. In [38], Khodayari *et al.* predicted the next probable location of the user by determining the speed and movement direction based on his/her last two recorded locations. Then, this prediction will be considered only when the localization result of WKNN [18] is substantially deviated from the prior location. The underline assumption is that users moving at both constant speed and direction, which is not the case in many practical scenes. In [39], Altintas *et al.* added a short term memory which stores the recent signal strength observations as the historical data. In the testing phase, the current RSSI readings and all historical RSSI readings in the memory are added and taken the average. This helps to eliminate the unexpected signal strength readings due to the reflection, diffraction and scattering of the radio waves. However, this method is valid only when the variation of RSSI between the current and previous positions is small, which is not always true.

In order to improve the localization stability, Xie *et al.* [22] used Spearman distance based on the RSSI ranking between APs. According to [22], although the absolute RSSI readings of a set of APs in a fixed location might be quite different, their rankings are more likely to remain the same, making it feasible to form a stable fingerprint. The drawback is that this algorithm is limited by the number of APs available. In the simulation of [22], there are 400 reference locations but only 4 APs which can provide a maximum of $4! = 24$ ranking fingerprints. Consequently, many different locations have the same fingerprints, leading to localization errors in the testing phase.

In general, all of the above methods provide acceptable accuracy within around twice the distance between two consecutive reference points (grid size), but the problems of KNN algorithm mentioned in Section 3.1 are not effectively solved. For

example, previous KNN research have not sufficiently investigated the inadequate sampling of RSSI due to the user's movement, i.e., only 1 or 2 RSSI readings are available in each testing location. Obviously this ignores a very important factor and affects the localization accuracy. In addition, in the methods that use historical data, the assumption that users moving in constant speed and direction is unrealistic in many scenes. Therefore, a new KNN algorithm, which addresses the aforementioned problems of KNN, is proposed.

2.3 System Model

2.3.1 Localization Scene

The fingerprinting localization system is generally divided into two phases: a training phase (offline phase) and a testing phase (online phase). In the training phase, features of the WiFi signals at each predefined reference point (RP) location, are collected and stored to a database. Those features typically include the mean and standard deviation of RSSI, the RSSI ratio between a pair of APs, the ranks of the APs, etc [2]. They individually or collectively form fingerprints at each RP. Here, we assume the area of interest has P APs and M RPs. For each RP i at its physical location $\boldsymbol{l}_i(x_i, y_i)$, a corresponding fingerprint vector is denoted as $\boldsymbol{f}_i = \{F_1^i, F_2^i, ..., F_N^i\}$, where N is the number of available features and $F_j^i (1 \leq j \leq N)$ is the j-th feature at point i. In the testing phase, each unknown location of the user, denoted as a testing point, is determined by the localization algorithm. During the training phase, multiple RSSI scans (S_1 scans) can be obtained at each location, and hence a set of RSSI values correspond to one RP while in the testing phase, only a small number of RSSI readings (S_2 scans), e.g., $S_2 = 1$ or $S_2 = 2$, is available for the fingerprint matching. Fig. 2.1(a) illustrates our localization scheme with 6 APs, 365 RPs and

175 testing points. Fig. 2.1(b) shows the heat map of 6 APs, where we represent signal strength by color. Clearly, the signals from 6 APs already cover the whole targeting area including 1 room and 3 corridors.

2.3.2 The classical KNN algorithm

The fingerprint distance between the unknown current point $\boldsymbol{l}$ and each reference point i in database is first calculated as follows

$$D_l^i = \sqrt{\sum_{j=1}^{N}(F_j - F_j^i)^2} \tag{2.1}$$

where F_j is the j-th fingerprint feature at the unknown location, N is the number of available fingerprints. Then K locations with the minimum distances are chosen as the K nearest neighbours. Finally, the position $\boldsymbol{l}$ of the user is determined by taking the average of all those K neighbours' locations.

2.3.3 Proposed Soft Range Limited KNN (SRL-KNN) method

SRL-KNN algorithm

This chapter proposes to leverage the information of the user's previous position, as the moving speed of a user is limited and one cannot instantaneously move to an unrealistic distant position from the prior one during the consecutive measurements. In a simple form, a circle can be drawn around the previous location to limit the nearest neighbour search space to within the circle, whose radius is determined by the user moving speed and time duration between two consecutive measurements. Instead of using that hard range limit, we here devise a novel soft range limiting factor to the fingerprint distance calculation where the locations near the user's previous

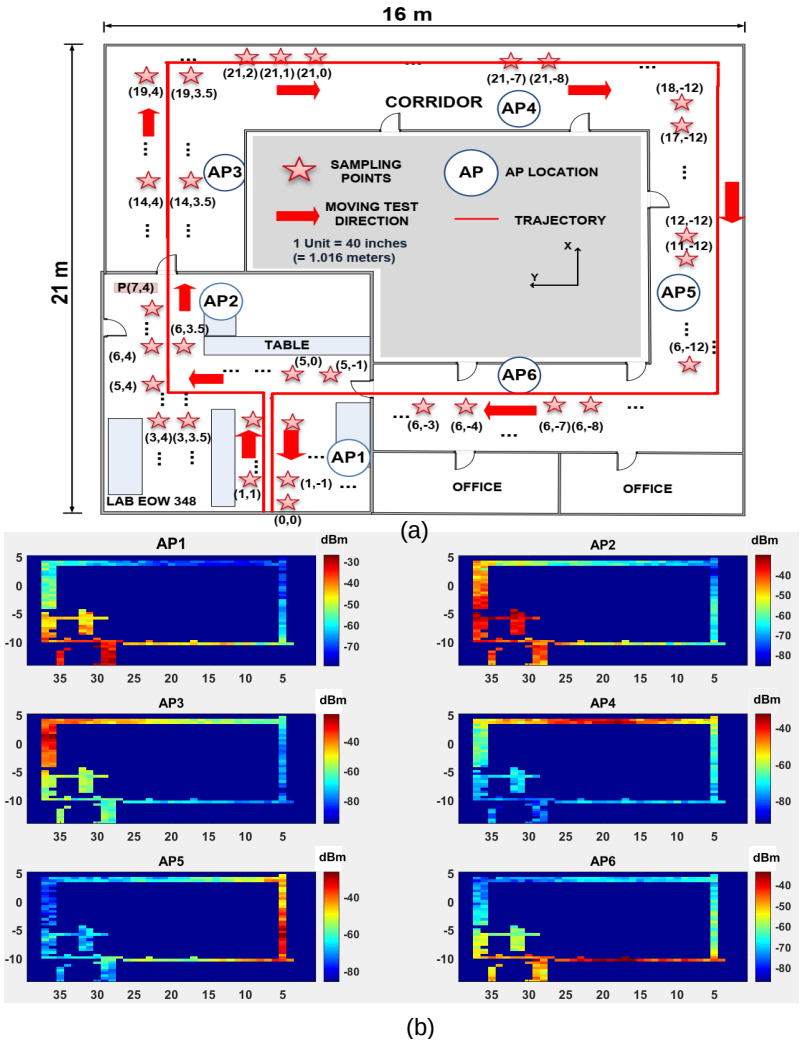


Figure 2.1: (a) Floor map of the test site. The solid red line is the mobile user's walking trajectory with red arrows pointing toward walking direction. (b) Heat map of the RSSI strength from 6 APs used in our localization scheme.

position are given higher likelihood to become one of K nearest neighbour candidates.

To achieve that, we modify the Euclidean distance in (2.1) as follows

$$\bar{D}_l^i = \frac{W_l^i \times D_l^i}{\sum_{i=1}^{M} W_l^i} \tag{2.2}$$

$$W_l^i = \exp\left(\frac{(x_i - x_{pre})^2 + (y_i - y_{pre})^2}{4\sigma^2}\right) \tag{2.3}$$

where W_l^i is the penalty function for the location i, M is the total number of RPs in the database, (x_{pre}, y_{pre}) is the most recent previous location of the user, σ is the maximum distance which the user can move in a consecutive sampling time interval Δt. For example, people tend to walk in indoor environments at a speed from 0.4 m/s to 2 m/s [40], [41] (maximum speed $v_{max} = 2$ m/s) and the user location will be updated every 1 second (consecutive sampling time interval $\Delta t = 1$ s). Therefore, $\sigma = v_{max}\ \Delta t = 2$ m. As shown in Fig. 2.2(c), the penalty function has the form of a Gaussian distribution with the mean being the previous location and the standard deviation being σ. Note that the prior position is only used here to form the soft range limit scaling factor as shown in (2.3), unlike in Kalman filter approaches which directly include the history position in the current location calculation. Moreover, our formulation only assumes a maximum moving speed, but does not require knowledge of the exact moving speed and direction of the user. The user's location $\boldsymbol{l}$ is determined through a weighted average of K nearest neighbours $\boldsymbol{l}_j$ as follows

$$\boldsymbol{l} = \frac{\sum_{j=1}^{K} \frac{\boldsymbol{l}_j}{\bar{D}_l^j}}{\sum_{j=1}^{K} \frac{1}{\bar{D}_l^j}} \tag{2.4}$$

where $\bar{D}_l^j$ is the modified Euclidean fingerprint distance which was presented in (2.2).

Fingerprint combination

In the WiFi fingerprinting method, the more stable the fingerprint is, the better the localization accuracy will be. However, the RSSI collected by a client device often experiences substantial fluctuations due to dynamically changing environments such as human blocking and movements, interference from other equipment and devices, receiver antenna orientation, etc., [28], [42]. Therefore, this chapter proposes to use the combination of a set of different fingerprints to ensure sufficient stability and

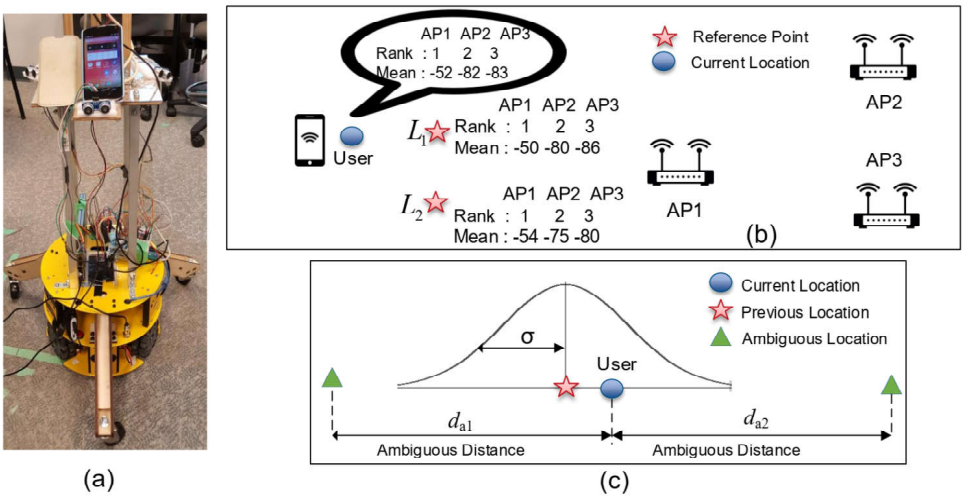


Figure 2.2: (a) 3-wheel robot. (b) Fingerprint combination illustration. (c) Penalty function illustration.

distinctive values in each location. The most common fingerprint used is the mean of RSSI [18], [21] which fluctuates significantly due to the previously mentioned effects. In contrast, one of the more reliable fingerprints is the mean difference of RSSI between a pair of APs. In [43], Dong *et al.* used two devices, i.e., a laptop and a smart phone to collect RSSI in a fixed location. They observed that although the individual RSSI readings of these devices fluctuate significantly, the mean differences of RSSI between pairs of APs are more stable. Therefore, the mean difference of RSSI can be used to address the received signal strength offset problem between different mobile devices. In addition, the rank fingerprints described in [22] can also be used as an additional fingerprint if there are enough number of APs available. Recently, Tian *et al.* [44] utilizes a new fingerprint named temporal correlation of the RSSI to improve the location estimation accuracy. However, in order to get the stable RSSI temporal correlation, a sufficient number of RSSI readings in each testing location is required, which is not feasible in our test cases. In our experiment, we first utilize some fingerprint types such as the RSSI differences and/or the AP rank to get n nearest neighbours RPs according to the shortest distance computed from (2.2).

Within the chosen nearest neighbours, we then refine our selection to K ($K < n$) nearest neighbours by using the mean of RSSI as the fingerprint. For example, Fig. 2.2(b) illustrates the scenario where we have a user trying to locate his location with the information of both the mean of RSSI and the rankings from 3 different APs. By using the rank fingerprints, two neighbours L_1 and L_2 are chosen based on the minimum fingerprint Euclidean distances. However, these points have the same rank fingerprints so we need to use the mean of RSSIs as the additional information to determine which point is the true neighbour of the user's location. With regard to mean fingerprints, neighbour L_1 that provides the smaller Euclidean distances is more likely the exact neighbour which we want to find.

Histogram of RSSI

As mentioned above, the raw RSSI readings at a location are unstable, fluctuating widely up to 10 dB [28]. Therefore, they may not represent well the feature of the RSSI at each location. In order to solve this problem, one may include the histogram of RSSI in the fingerprint distance calculation, which defines the probability of the original RSSI reading of the jth AP falling into $[R_j - 0.5$ dBm, $R_j + 0.5$ dBm$]$ at the reference location i as follows [45]

$$p_R^{i,j} = \frac{n_{R_j}^i}{n_{total}^{i,j}} \tag{2.5}$$

where $n_{total}^{i,j}$ is the total number of RSSI scans of the jth AP at location i, $n_{R_j}^i$ is the number of RSSI readings of the jth AP falling into the range between $R_j - 0.5$ dBm and $R_j + 0.5$ dBm ($R_L^j \leq R_j \leq R_U^j$), R_L^j and R_U^j are the minimum and maximum values of RSSI of jth AP respectively. Consequently, (2.1) can be modified as a

weighted distance according to

$$D^i_{l,hist} = \sqrt{\sum_{j=1}^{N} \sum_{R_j=R^j_L}^{R^j_U} p^{i,j}_R (F_j - R_j)^2} \tag{2.6}$$

and the final fingerprint distance is obtained as

$$\bar{D}^i_l = \frac{W^i_l \times D^i_{l,hist}}{\sum_{i=1}^{M} W^i_l} \tag{2.7}$$

2.4 Experiment And Analysis

2.4.1 Experimental Setup

All experiments have been carried out on the third floor of Engineering Office Wing (EOW), University of Victoria, BC, Canada. The dimension of the area is 21 m by 16 m. It also has 3 long corridors as shown in Fig. 2.1(a). The RSSI measurements were taken in 365 pre-determined RPs. A mobile device (Google Nexus 4 running Android 4.4) mounted on a 3-wheel robot (Fig. 2.2(a)) was sent to target locations to collect fingerprints. The localization accuracy of the robot is 0.07 m $\pm$ 0.02 m. At each location, 100 instantaneous RSSI measurements ($S_1 = 100$) were collected to a database. There are 6 APs and 5 of them provide 2 distinct MAC address for 2.4 GHz and 5 GHz communication channels respectively. Equivalently, in every scan, 11 RSSI readings from those 6 APs can be collected.

In the testing phase, we conducted both one-location test and trajectory test. In the one-location test, RSSI values at a fixed position were collected and the user's position was determined in every consecutive sampling time interval Δt. In the trajectory test, the robot carried a mobile device and moved along the direction as shown by the red solid line in Fig. 2.1(a). RSSI readings were collected continuously by the

Table 2.1: Average localization errors

Method	**SRL-KNN Mean**	**SRL-KNN Rank**
Average Error (m)	0.81 ± 0.40	1.20 ± 0.96
Method	**SRL-KNN Histogram**	**SRL-KNN Mean and Rank**
Average Error (m)	0.66 ± 0.36	0.76 ± 0.51
Method	**SRL-KNN Mean-RSSI Differences**	**RADAR [21]**
Average Error (m)	0.71 ± 0.46	1.19 ± 0.86
Method	**STI-WKNN [35]**	**Spearman Rank [22]**
Average Error (m)	1.09 ± 0.81	1.45 ± 1.14
Method	**Kernel Method [46]**	**Kalman Filter [24]**
Average Error (m)	1.07 ± 0.86	0.96 ± 0.48

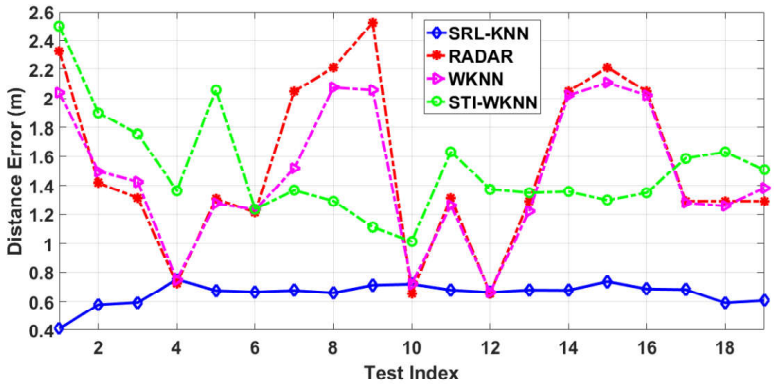


Figure 2.3: Localization errors of one-location test.

phone and were transmitted to a server in real time. The server analyzed the data to locate the user's position. The mean fingerprint in each location was determined by the average of S_1 RSSI readings for training and S_2 RSSI readings for testing. On the other hand, the mean difference of RSSI fingerprint for a test location was calculated by taking the average of S_1 (S_2) RSSI differences between a pair of APs.

2.4.2 One-Location Test

In this test, the mobile device was put on the location $P(7, 4)$ as shown in Fig. 2.1(a). The experiment was conducted in busy hours when many students (up to 10) used WiFi and moved around the lab. A maximum RSSI standard deviation of 5.5 dB was recorded over 100 consecutive RSSI readings. The large fluctuation of RSSI is due to the factors explained in Subsection. 2.3.3.

Fig. 2.3 shows the comparison of the localization accuracy among the classical KNN fingerprinting algorithms in RADAR [21], WKNN [18], STI-WKNN [35] and our proposed SRL-KNN algorithm. All algorithms use the mean of RSSI as the fingerprint, the consecutive sampling time interval $\Delta t = 1$ s and the number of nearest neighbours $K = 3$. The user location is estimated based on 1 RSSI scan ($S_2 = 1$) collected every Δt. Over all 19 tests conducted at different time instants within one hour, the localization results of RADAR, WKNN and STI-WKNN fluctuate more than 1.7 m from 0.70 to over 2.40 m, while SRL-KNN reports a much lower fluctuation with 0.3 m from 0.40 to 0.70 m. The accuracy of SRL-KNN is 2 times better than the other methods with average distance error being 0.60 m compared with over 1.20 m of the others.

2.4.3 Trajectory Test

In this test, the robot moved along a pre-defined route as shown in Fig. 2.1(a) with an average speed around 0.6 m/s. All the testing locations (total 175 locations) along the trajectories are randomly picked. In this experiment, the maximum speed in our algorithm is set to $v_{max} = 2$ m/s, so the maximum distance which user can move is $\sigma = v_{max} \times \Delta t = 2$ m. The initial position of the user in these testing trajectories is assumed to be known. All the other parameters are the same as those in the one-location test.

Fig. 2.4(a) compares the cumulative distribution function (CDF) of localization errors between SRL-KNN and other KNN methods, i.e., RADAR [21], Spearman rank distance [22], STI-WKNN [35]. Here, for comparison, we used both mean of RSSI, rank of APs as our fingerprints. Clearly, the SRL-KNN (blue line) outperforms the other methods in terms of positioning accuracy. Further analysis shows that due to larger RSSI fluctuations, the other methods may choose a wrong location with

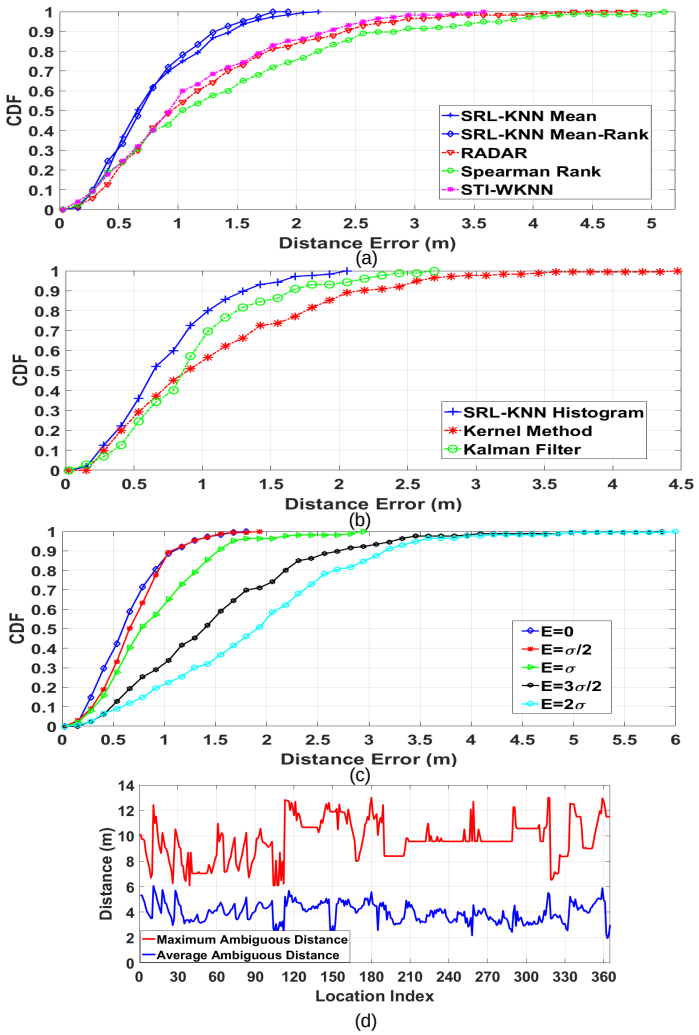


Figure 2.4: (a) CDF of localization errors of SRL-KNN using mean and rank database and other KNN methods. (b) CDF of localization errors of SRL-KNN using histogram and other probabilistic methods. (c) CDF of localization errors of SRL-KNN using histogram in different error scenarios of historical data. (d) Maximum and average ambiguous distances of 365 locations in the database

similar fingerprints as its nearest neighbours. Note that such location could be far from the actual location, leading to an extreme large error in the scale of the testing site dimension. As shown in Fig. 2.4(a), a 4.80 m maximum localization error is recorded for RADAR, 3.50 m for STI-WKNN and the largest maximum localization error of over 5 m for Spearman rank method. In contrast, SRL-KNN eliminates

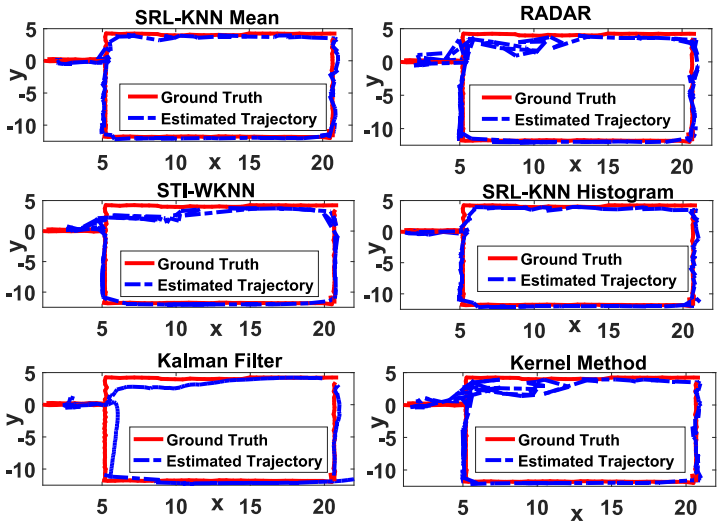


Figure 2.5: Ground truth and estimated trajectories. Red line represents the trajectory ground truth. Blue lines are estimated trajectories

such error pattern, resulting in a much smaller maximum errors of 2.20 m with the mean fingerprint. In particular, SRL-KNN using only mean fingerprint has 80% of the location error within 1.20 m while RADAR, STI-WKNN and Spearman rank distance are 1.80 m, 1.80 m and 2.30 m respectively. To achieve higher accuracy, the combination of different fingerprint described in Subsection 2.3.3 is used. In this article, we implemented two different fingerprint combinations: use the mean RSSI with the rank fingerprint and use the mean RSSI with RSSI difference between a pair of APs. In both cases, the rank or RSSI difference fingerprint is firstly utilized to get $n = 7$ neighbours and then $K = 3$ refined nearest neighbours are chosen based on the mean fingerprint. These two methods have the similar performance with the maximum error of around 1.80 m and 80% of the error is within 1 m.

We further implement the histogram based fingerprint distance described in Subsection 2.3.3. In the testing phase, the feature F_j in (2.6) is obtained as the mean of all S_2 RSSI readings from an AP. In comparison with the other probabilistic approaches including Kernel method [46], Kalman filter [24] in Fig. 2.4(b), this approach clearly

Table 2.2: Average localization errors of UJIIndoorLoc database

	SRL-KNN Mean	**RADAR [21]**	**STI-WKNN [35]**
Building 0 (m)	4.7 ± 2.7	7.9 ± 4.9	7.9 ± 5.2
Building 1 (m)	4.6 ± 3.8	8.2 ± 4.9	6.8 ± 6.1
Building 2 (m)	6.0 ± 4.5	8.2 ± 7.4	6.1 ± 3.7
All buildings (m)	5.0 ± 3.7	7.7 ± 6.0	7.0 ± 4.9

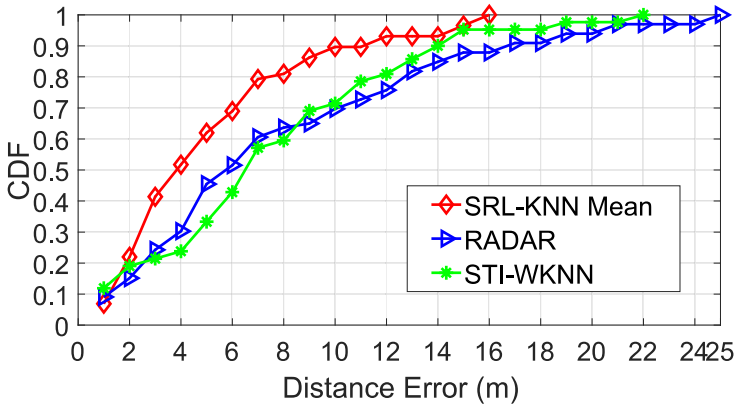


Figure 2.6: CDF of localization errors of UJIIndoorLoc database for all 3 buildings

outperforms. Our method (plus markers) has a maximum error of 2.10 m while the maximum errors of Kalman filter (circle markers) and Kernel method (star marker) are 2.70 m and 4.50 m, respectively. The 80% of the error in our histogram approach is 0.90 m, following by 1.40 m of Kalman filter and 1.90 m of Kernel method. Fig. 2.5 illustrates the ground truth and estimated trajectory using different methods. As clearly shown, both histogram and mean fingerprint SRL-KNN are the most accurate predictions. In addition, Table 3.1 lists all the average localization errors. The best accuracy is 0.66 m in the case of SRL-KNN using the RSSI histogram. Regarding the computational complexity, SRL-KNN has the similar complexity $O(MN)$ with the conventional KNN RADAR [21], where M is the number of RPs, and N is the number of available features.

Since SRL-KNN leverages the information of a user's previous position to estimate the current location, the performance of SRL-KNN depends on the accuracy

of historical data. Note that all of our SRL-KNN results presented so far are based on the estimated imperfect history data. In order to look into the propagation error due to the imperfect prior location estimation, Fig. 2.4(c) illustrates the localization errors of SRL-KNN using histogram based fingerprint distance with both the ideal and erroneous history data. Starting with the perfect historical coordinate $\boldsymbol{h}(x, y)$ for every location in the testing trajectories mentioned in Sec. 2.4.3, an amount of error E m is added to $\boldsymbol{h}$. The erroneous prior location $\boldsymbol{h'}(x', y')$ is obtained as: $x' = x + x_e$, $y' = y + y_e$, where x_e and y_e are random variables that follow Gaussian distribution

$$x_e \sim \mathcal{N}(0, \sigma_{x_e}^2)\,;\; y_e \sim \mathcal{N}(0, \sigma_{y_e}^2)\,;\; \sqrt{\sigma_{x_e}^2 + \sigma_{y_e}^2} = E$$

Fig. 2.4(c) shows the cases where E is proportional to $\sigma = 2$ m. Obviously, if the error E of the history data is within $\sigma/2$ m, the localization accuracy is mostly similar to the ideal case, with a maximum error of 1.90 m and 80% of the error is 1 m. When E increases to σ m, the accuracy becomes slightly worse with the maximum error being 2.90 m and 80% error being around 1.50 m. As shown in Table 3.1, all of the average errors of SRL-KNN are around $\sigma/2$, which indicates that SRL-KNN is robust to localization error of the previous position. If the value of error E is larger than σ, i.e., $E = 3\sigma/2$ or $E = 2\sigma$, the performance will degrade and the accumulated errors become more significant. The theoretical explanation is as follows. SRL-KNN implements a penalty function based on the previous location to discriminate the ambiguous locations. A location $\boldsymbol{l}_j$ is defined as an ambiguous point of $\boldsymbol{l}_i$ if their physical distance is larger than the grid size but their two vectors $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ have a fairly high Pearson correlation coefficient above the correlation threshold. We choose the value of the correlation threshold equal to the average correlation coefficients between $\boldsymbol{l}_i$ and all of its physical nearest neighbours, i.e., approximately 0.85 in our database. Then all non-nearest-neighbour locations whose correlation

coefficient above this threshold are considered as ambiguous points. Note that two locations are defined as physical neighbours if the physical distance between them within the grid size. The ambiguous distance d_a is defined as the physical distance between a location and its ambiguous point. Pearson correlation coefficient $\rho(\boldsymbol{f}_i, \boldsymbol{f}_j)$ between $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ can be calculated as follows

$$\rho(\boldsymbol{f}_i, \boldsymbol{f}_j) = \frac{1}{N-1} \sum_{n=1}^{N} (\frac{F_n^i - \mu_i}{\delta_i})(\frac{F_n^j - \mu_j}{\delta_j})$$

where N is the number of available fingerprints, μ_i, μ_j are the means of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ respectively, δ_i, δ_j are the standard deviations of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ respectively. Fig. 2.2(c) shows that if the error of previous location is within $d_a - \sigma$, the penalty function can provide higher likelihood to the potential locations near the correct current position and lower likelihood to the other ambiguous locations. Therefore, the estimation accuracy of the current location will not be adversely affected. In order to estimate d_a, Fig. 2.4(d) illustrates the maximum and average ambiguous distances of all 365 locations in the database. The average ambiguous distance $\bar{d_a}$ is around 4 m (2σ) and the maximum value d_a^{max} is above 12 m (6σ). These results affirm that if the error of the previous locations is within $\bar{d_a} - \sigma = \sigma$, SRL-KNN is robust to the localization error of the previous position. Furthermore, according to the survey in [47], the percentage of stationary time can exceed 80% for most mobile users. During the no movement period, the number of RSSI readings collected in one-location (S_2) is sufficient to improve the conventional KNN accuracy. Therefore, in order to enhance the accuracy when locating a user's position in a long trajectory, we can employ these stationary locations as aligning points where the prior locations can be ignored. In that case, some classical KNN approaches including RADAR [21], WKNN [18] or STI-WKNN [35] can be exploited to estimate the user's location.

In order to prove the consistent effectiveness of SRL-KNN, our algorithm is implemented with another published dataset, namely UJIIndoorLoc [48]. The reported average localization error in [48] is 7.9 m. The training and validation data in all 3 buildings of the databse from 2 random phone users (Phone Id: 13, 14) are used to implement SRL-KNN algorithm. The maximum distance between 2 consecutive locations in the testing trajectory can be up to 20 m so $\sigma = 20$ m is chosen. Note that the grid size of UJIIndoorLoc is different from our collected database so the average localization error for UJIIndoorLoc is different from that reported previously. However, the relative accuracy comparison between SRL-KNN and conventional KNN, e.g., RADAR [21] or STI-WKNN [35] can still reflect well the effectiveness of our algorithm. Table 2.2 shows the average errors in meter of SRL-KNN, RADAR, STI-WKNN for each separate building and for all 3 buildings in general. These results consistently illustrate that SRL-KNN is more robust than other conventional KNN algorithms including RADAR [21] and STI-WKNN [35]. For all 3 buildings, the average error of SRL-KNN using mean fingerprint is 5.0 m while the result of RADAR is 7.7 m and STI-WKNN is 7.0 m. Furthermore, Fig. 2.6 compares the CDF of localization errors between 3 methods. In total, a 16 m maximum localization error is recorded for SRL-KNN, 22 m for STI-WKNN and the largest maximum localization error of 25 m for RADAR. Besides, 80% of the error is below 7 m in the case of SRL-KNN, which is much lower than 13 m and 12 m in the case of RADAR and STI-WKNN, respectively.

2.5 Conclusions

In conclusion, we have proposed a low complexity soft range limited KNN (SRL-KNN) for WiFi indoor localization. This algorithm exploits the information of previous

positions and simultaneously applies the soft range limiting factor for fingerprint distance calculation to achieve more accurate and stable positioning performance. We demonstrated that SRL-KNN can address effectively some main challenges of KNN including the spatial ambiguity, RSSI instability and the RSSI short collecting time, especially when RSSI histogram is taken into account in calculating fingerprint distance. Experimental results have shown that SRL-KNN achieves the best accuracy of 0.66 m with 80% of the error within 0.89 m, which outperforms existing KNN methods.

Chapter 3

Semi-Sequential Probabilistic Model For Indoor Localization Enhancement

3.1 Introduction

As mentioned in Chapter 2, probabilistic methods are based on the statistical inference between the measured signal and the stored fingerprints through Bayes rule [23]. They have medium complexity and provide good accuracy in indoor localization [2, 49]. Therefore, probabilistic methods are widely utilized to extract information from both RSSI and CSI fingerprints. In order to determine the location, probabilistic methods require the probability density function (PDF) of the fingerprinting features. Some early research assume the RSSI PDF follows empirical parametric distributions such as single-peak [50] or double-peak Gaussian [51], lognormal distribution [52], etc. Ref. [50] observes experimentally that most RSSI values follow single-peak Gaussian distribution and tend to be left-skewed. The left-skewed distribution occurs when

the variation of the weaker RSSI is larger than that of the stronger one. In a later research, Ref. [52] indicates that the RSSI distribution can be left-skewed, symmetric or right-skewed depending on the distance and obstacles between the user and access points (AP). Therefore, the lognormal PDF is more suitable to approximate the right- or left-skewed RSS distribution. Although a common assumption about the RSS distribution is single-peak Gaussian, Ref. [51] claims that RSSI can be double-peak Gaussian distribution (DGD) under some circumstances and suggested to replace the single-peak Gaussian distribution with DGD. However, all of these observations are highly dependant on the experiment specifics and may not be reasonable approximations [53].

To achieve better performance by eliminating the assumption on the RSSI PDF, [54] exploits both histogram and Kernel methods to estimate the PDF. In contrast, such methods are called non-parametric. Here, the histogram of RSSI is estimated for each AP with a set of non-overlapping bins that cover the whole range of the RSSIs for each AP. Then the RSSI PDF is calculated as a piecewise constant function where the density is constant within each bin. In [46], the experiment by Kushki *et al.* shows that a bin width of 10 dB provides the lowest positioning error. Works in Horus [23] compare the performance between parametric and histogram methods. It is found that both of those methods provide comparable performances with a slight advantage for the parametric one. The reason is from the existence of zero count bins in the histogram due to the limited number of different signal strength values in the training phase. As an improvement to the histogram method, Kernel method employs component smoothing functions for each data value to produce a smooth and continuous probability curve and avoid any zero count bins. Ref. [55] further proposes a non-parametric statistical model with Parzen window density estimation. The kernel for Parzen window needs to be non-negative and normalized. Among all of

the suitable kernels such as Epanechnikov, logistic and sigmoid, etc., Gaussian kernel is claimed to have consistently good performance and is the most widely used.

In order to construct a fingerprint map for accurate localization, a large number of RPs are required [9], which is time-consuming and labor-intensive [1]. Recently, probabilistic Gaussian process (GP) is utilized to enhance the accuracy of indoor localization in the uncalibrated domain with a limited number of RPs. GP is another non-parametric model characterized completely by its mean function and co-variance matrix. Ref. [56] presents GP regression models to predict the spatial distribution of RSSI. The appropriate compound kernel functions are systematically selected instead of a single kernel function to get the heterogeneous RSSI PDF. In Ref. [57], GP is trained by the firefly algorithm (FA) to obtain the hyper-parameters. Once the GP hyper-parameters are obtained, the GP prior distribution can be used for regression to predict RSSI at locations with no prior measurements.

Besides RSSI, CSI fingerprints are widely used in probabilistic methods. FILA [58], one of the early works using CSI, estimates the signal strength distribution for each AP at each location based on the total power of all CSI sub-carriers. After the CSI power distribution database is constructed, the probabilistic method with Bayes' rule predicts the user's location. Later research on BiLoc [7] exploits bi-modal data that estimates the angle of arrival (AoA) and average amplitudes of CSI as the fingerprints for localization. The advantage of the method is that when there is no line of sight (LoS), the CSI amplitude will be significantly reduced but AoA will be less affected. In contrast, when LoS is available, the CSI amplitude is a stable fingerprint to be relied on. In the testing phase, the probabilistic approach is adopted to localize the user's position using those bi-modal fingerprints.

Table 3.1 summarizes the experiment specifics and results of typical probabilistic methods. Here the number of APs, RPs and testing points vary among experiments

Table 3.1: Comparisons of Indoor Localization Experiments Using Probabilistic Techniques

Method	Feature	Access point (AP)	Reference point (RP)	Testing Point	Grid Size	Accuracy
DGD [51]	RSSI	-	68	35	2.5 m	2.8 m
Horus [23]	RSSI	21	172	100	1.52 m	0.42 ± 0.28 m
Kernel method [46]	RSSI	33	66	44	2 m	2.31 ± 2.10 m
BiLoc [7]	CSI	1	15	15	1.8 m	1.5 ± 0.8 m
FILA [58]	CSI	1-3	28	-	-	0.4 m to 1 m

and the grid size is defined as the distance between two consecutive RPs.

In general, probabilistic methods provide acceptable accuracy from 1 m to 2.5 m. However, all of the above methods treat all the locations in the database with equal probability in predicting the current location, which ignores its correlation to the user's previous position. Since the moving speed of the user in an indoor environment is bounded, the locations near the previous one should have higher probability to be estimated as the user's current point than others. Therefore, in this chapter, we propose a simple short term memory step for all existing probabilistic methods to enhance their performances. Our semi-sequential probabilistic model (SSP) applies window functions such as Gaussian, Hann and Tukey, and is based on the physical distance between the RP and the user's predicted previous position to calculate the probability of RP being near the user's current position. As a result, the spatial ambiguity of fingerprints [8] is significantly reduced and the localization accuracy is improved.

In the literature, the idea of exploiting the measurements in previous time steps to locate the current location was adopted in the research of recurrent neural network (RNN) [59], Kalman filter [24–27] and soft range limited K-nearest neighbors (SRL-KNN) [8]. In Ref. [59], the proposed P-MIMO LSTM model exploits the sequential RSSI measurements and the trajectory information from multiple time steps to achieve high accuracy. However, the complexity is high, and the long term memory dependency can cause the significant accumulated errors if the inaccuracy in histor-

ical data is high. On the other hand, Kalman filter [24] estimates the most likely current location based on prior measurements and Gaussian noises with linear motion dynamics assumptions. In real scenarios, those assumptions are not necessarily valid [28]. SRL-KNN [8] does not require the above assumptions and reaches the lowest complexity. However, the modified penalty functions in SRL-KNN can only be applied for Euclidean distance, not for probabilistic model. In contrast to those approaches, SSP is able to boost the performance of several probabilistic systems using RSSI or CSI fingerprints. Furthermore, the short term memory dependency ensures our low complexity and avoid accumulating errors. The proposed model is tested with several experiments using both RSSI and CSI fingerprints and compared with existing probabilistic methods.

3.2 Proposed Model

3.2.1 Proposed Localization System

The proposed localization system has two phases: a training phase and a testing phase. In the training phase, fingerprints at each predefined reference point (RP) location are collected and stored in a database. The fingerprints can be either RSSI or CSI. We assume the area of interest has P APs and M RPs. For each RP i at its physical location $\boldsymbol{l}_i(x_i, y_i)$, a corresponding fingerprint vector is denoted as $\boldsymbol{F}(\boldsymbol{l}_i) = \{F_1(\boldsymbol{l}_i), F_2(\boldsymbol{l}_i), ..., F_N(\boldsymbol{l}_i)\}$, where N is the number of available features and $F_j(\boldsymbol{l}_i)$, $1 \leq j \leq N$, is the j-th feature at point i. In the testing phase, each unknown location of the user, denoted as a testing point, is determined by the localization algorithm.

3.2.2 Proposed Semi-Sequential Probabilistic (SSP) Model

Conventional Probabilistic Method

In the testing phase, we assume that a fingerprint vector $\boldsymbol{F}(\boldsymbol{l}_{curr})$ at the unknown current location $\boldsymbol{l}_{curr}$ is measured. The likelihood function $P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)$ describes the probability of the k-th fingerprint feature at location $\boldsymbol{l}_i$, has a signal strength $F_k(\boldsymbol{l}_i) \approx F_k(\boldsymbol{l}_{curr})$ condition on $\boldsymbol{l}_{curr} \approx \boldsymbol{l}_i$, which can be estimated by the parametric [50–52] or non-parametric [23, 54] methods. Therefore, the probability of the current location $\boldsymbol{l}_{curr}$ being close to $\boldsymbol{l}_i$ given the measured fingerprint vector $\boldsymbol{F}(\boldsymbol{l}_{curr})$ can be estimated according to Bayes' theorem [2]. Here, $\boldsymbol{l}_{curr}$ will fall into the set of $\boldsymbol{l}_i$.

$$P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr})) = \prod_{k=1}^{N} \frac{P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)P(\boldsymbol{l}_i)}{P(F_k(\boldsymbol{l}_{curr}))} \tag{3.1}$$

Here, Eq. (3.1) is valid following the assumptions along with other researchers that F_k are mutually independent, which leads to $P(\boldsymbol{F}(\boldsymbol{l}_{curr})|\boldsymbol{l}_i) = \prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)$ and $P(\boldsymbol{F}(\boldsymbol{l}_{curr})) = \prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr}))$. Each location $\boldsymbol{l}_i$ has a priori probability $P(\boldsymbol{l}_i) \equiv P(\boldsymbol{l}_i \approx \boldsymbol{l}_{curr})$ to be nearest to $\boldsymbol{l}_{curr}$, which is initially assumed to be equally likely for every location in the database. Furthermore, $\prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr}))$ is independent of $\boldsymbol{l}_i$. Therefore, for simplification, Eq. (3.1) can be modified to

$$P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr})) \propto \prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i) \tag{3.2}$$

Eq. (3.2) is applied for the whole database to get the set of $P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr}))$ including all of the RPs. The maximum value of $P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr}))$ can be chosen as the user's location [23]. In some approaches [51], K biggest values are chosen, and the user's location is the average of all K values.

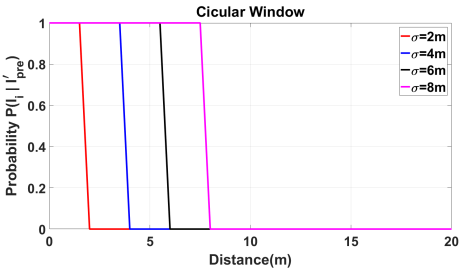


Figure 3.1: (a) Circular Window

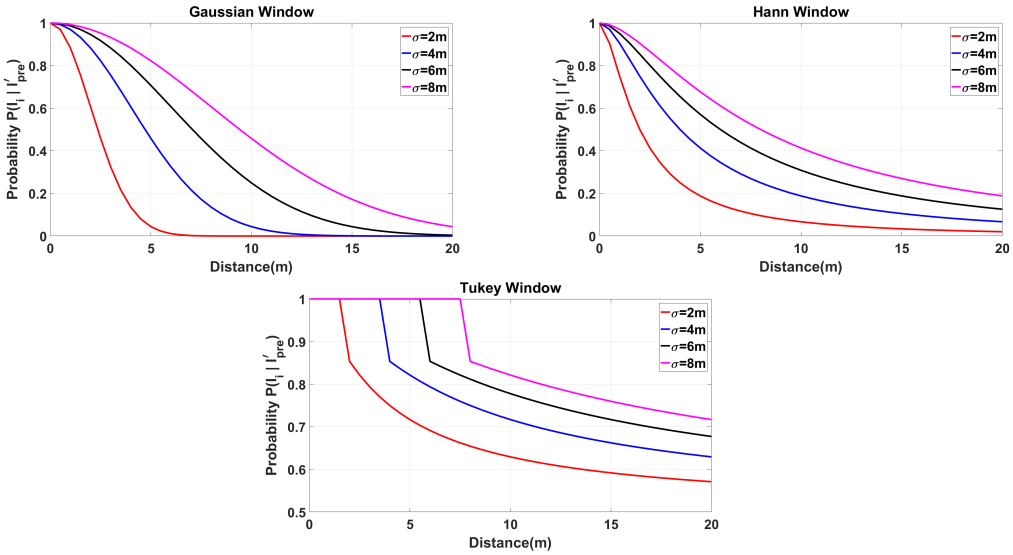


Figure 3.2: Short term memory window forms with different values of σ. (a) Circular Window. (b) Gaussian Window. (c) Hann Window. (d) Tukey Window.

Proposed SSP Model

As the moving speed of a user is bounded, it is highly unlikely for a user to move to an unrealistic distant position from the prior one during the consecutive measurements. Therefore, the assumption that $P(\boldsymbol{l}_i)$ is equally likely for every location is not valid. In fact, the locations near the previous one should have higher probability than the others. In order to incorporate that into the probabilistic models, a short term memory window is proposed to determine the possibility of a reference location $\boldsymbol{l}_i$ to be the nearest to the current location $\boldsymbol{l}_{curr}$. For example, a circular window whose radius is determined by the user moving speed and time duration between two consecutive

measurements can be drawn around the previous location to limit the nearest neighbor search within the area. Besides the hard range limit, we here propose several soft range windows based on the predicted previous location $\boldsymbol{l}'_{pre}(x'_{pre}, y'_{pre})$:

– Gaussian window:

$$P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre}) = \exp(-\frac{D^2}{2\sigma^2}) \tag{3.3}$$

– Hann window:

$$P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre}) = \cos^2\left[\frac{\pi D}{2\cdot(D+\sigma)}\right] \tag{3.4}$$

– Tukey window:

$$P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre}) = \begin{cases} 1, & \text{if } D < \sigma \\ \frac{1}{2}\left\{1+\cos\left[\frac{\pi D}{2\cdot(D+\sigma)}\right]\right\}, & \text{if } D \geq \sigma \end{cases} \tag{3.5}$$

where $D(\boldsymbol{l}_i, \boldsymbol{l}'_{pre}) = \sqrt{(x_i - x'_{pre})^2 + (y_i - y'_{pre})^2}$ is the distance between $\boldsymbol{l}_i$ and $\boldsymbol{l}'_{pre}$. The standard deviation σ determines the spread of the window. Fig. 3.2 illustrates the shapes of the above windows and the dependency between the probability of the considered location in the database $P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})$ and the distance D. Clearly, $P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})$ is inversely proportional to D. Here, σ can be determined according to the maximum distance d_{max} that a user can move during the sampling time interval Δt. As shown in Fig. 3.2, the larger the σ is, the more spread out the short term memory window becomes.

After applying the short term memory window for the whole database, the probability $P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})$ can be normalized as follows.

$$\hat{P}(\boldsymbol{l}_i|\boldsymbol{l}'_{pre}) = \frac{P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})}{\sum_{i=1}^{M} P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})} \tag{3.6}$$

For simplicity, we denote $\hat{P}(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})$ as $P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})$ in the rest of the chapter unless otherwise specified. Finally, Eq. (3.1) is modified with the above short term memory step as follows.

$$P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr}), \boldsymbol{l}'_{pre}) = \prod_{k=1}^{N} \frac{P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i, \boldsymbol{l}'_{pre})P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre})}{P(F_k(\boldsymbol{l}_{curr}))} \tag{3.7}$$

$\prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr}))$ can still be ignored here due to the same argument in Subsection 3.2.2. As $P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i, \boldsymbol{l}'_{pre})$ is independent of the previous predicted location $\boldsymbol{l}'_{pre}$, i.e. $P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i, \boldsymbol{l}'_{pre}) = P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)$, Eq. (3.7) becomes

$$P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr}), \boldsymbol{l}'_{pre}) \propto \prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)P(\boldsymbol{l}_i|\boldsymbol{l}'_{pre}) \tag{3.8}$$

With this simple short term memory, we can convert the existing memoryless probabilistic methods, e.g., Horus [23], DGD [51], Kernel method [46], FILA [58], BiLoc [7], to semi-sequential model and improve their performances significantly. After having $P(\boldsymbol{l}_{curr}|\boldsymbol{F}(\boldsymbol{l}_{curr}), \boldsymbol{l}'_{pre})$, the maximum value of that probability can be chosen as the user's location [23,46]. On the other hand, in [51], K biggest values are chosen, and the user's location is the average of all K values.

3.3 Database And Experiments

This chapter investigates both RSSI and CSI fingerprints for localization. All experiments have been carried out on the third floor of Engineering Office Wing (EOW), University of Victoria, BC, Canada. The dimension of the area is 21 m by 16 m. It has three long corridors as shown in Fig. 3.3(a). The RSSI and CSI fingerprints for both training and testing are collected using an autonomous driving robot. The

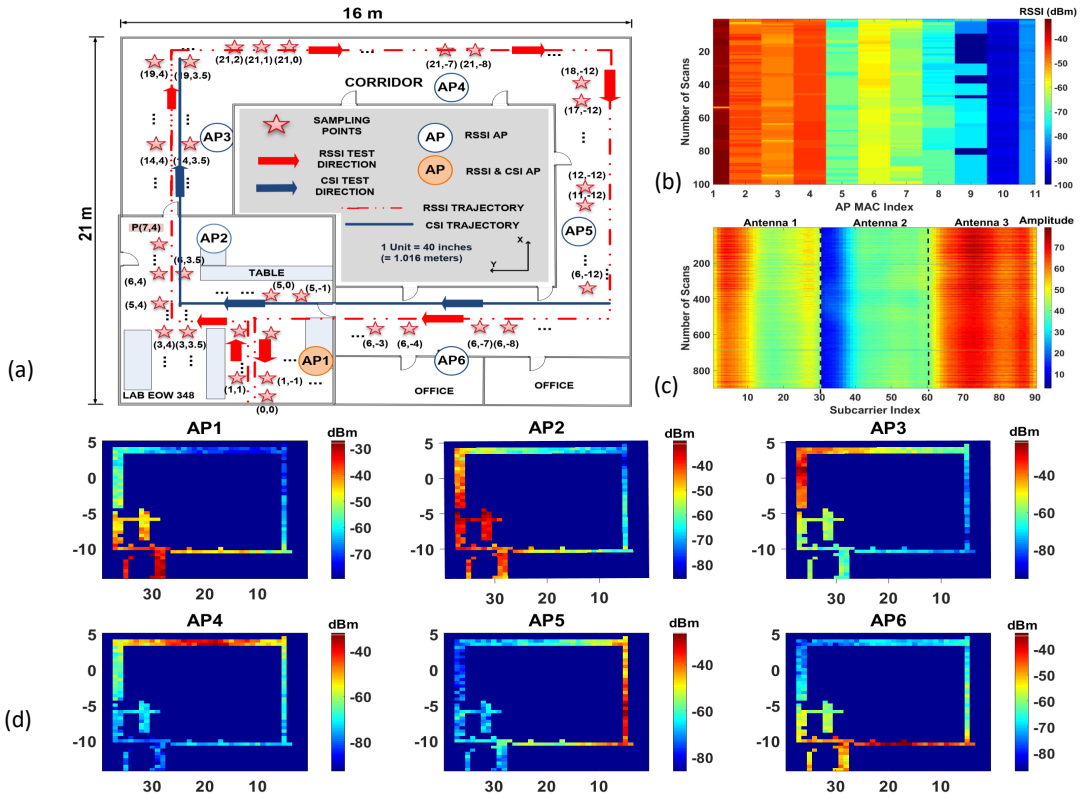


Figure 3.3: (a) Floor map of the RSSI and CSI test site. The solid red line is the RSSI trajectory with red arrows pointing toward walking direction. The dash green line is the CSI trajectory. (b) An example of a collected RSSI vector with 100 scans. (c) An example of a collected CSI image from Intel 5300 NIC. (d) Heat map of the RSSI strength from 6 APs used in our localization scheme.

3-wheel robot has multiple sensors including a wheel odometer, an inertial measurement unit (IMU), a LIDAR, 3 sonar sensors and a color and depth (RGB-D) camera. It can navigate to a target location within an accuracy of 0.07±0.02 m.

During RSSI localization experiments, the robot carried a mobile device (Google Nexus 4 running Android 4.4) to collect RSSI. In the training phase, 100 RSSI scans ($S_1 = 100$) were obtained at each location. Fig. 3.3(b) illustrates an instance of the 100-scan RSSI from 11 AP network interface controllers (NICs). In this experiment, there are 6 APs and 5 of them have 2 distinct MAC address each assigned to two associating NICs with 2.4 GHz and 5 GHz communications channels respectively,

except for one that operates on 2.4 GHz frequency only. In total, each scan contains 11 RSSI readings. In the testing phase, the user location will be updated at a consecutive sampling time interval $\Delta t = 1$ s. During that time interval, only a small number of RSSI scans (S_2 =1 or 2) were available for the fingerprint matching. Fig. 3.3(a) illustrates our localization scheme with 365 RPs (pink stars) and 175 testing points (dash red line). Fig. 3.3(d) shows the heat map of 6 APs, where the signal strength is represented by color. Clearly, the signals from 6 APs cover the whole targeting area including 1 room and 3 corridors.

In the same targeting area, only one single AP (AP1) that has 3 receiving antennas with 30 multiple subcarriers each, was set up for CSI measurements. The robot carrying an Intel WiFi Link 5300 MIMO NIC navigated across 145 RPs and 60 testing points (along the blue solid line in Fig. 3.3(a)) to collect CSI with csitool [60]. A single CSI data is a complex number to represent its amplitude and phase. However, the phase of CSI is noisy due to effects such as fading and frequency fluctuation [5, 61]. Therefore, only CSI amplitude is utilized in this chapter. In each measurement at RP i, we group H number of CSI measurements with W subcarriers to construct an $H \times W$ matrix

$$\tilde{A}(\boldsymbol{l}_i) = \begin{bmatrix} \tilde{A}_i^{11} & \tilde{A}_i^{12} & \dots & \tilde{A}_i^{1W} \\ \tilde{A}_i^{21} & \tilde{A}_i^{22} & \dots & \tilde{A}_i^{2W} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_i^{H1} & \tilde{A}_i^{H2} & \dots & \tilde{A}_i^{HW} \end{bmatrix}. \tag{3.9}$$

Here, $\tilde{A}_i^{hw}$ is the CSI amplitude value of the subcarrier w in the measurement h at RP i. Fig. 3.3(c) shows an image representation of collected CSI with $H = 900$ and $W = 90$.

In the testing phase, the robot moved along a pre-defined route (Fig. 3.3(a)) at a speed randomly changing within (0.6-4.0) m/s to simulate the indoor walking pattern

Table 3.2: Average Errors (SSP Model with Gaussian Window, $\sigma = d_{max}$)

Method	Fingerprint	AP Number	Original Model (m)	SSP Model (m)
Horus	RSSI	6	1.5 ± 1.2	1.0 ± 0.7
DGD	RSSI	6	1.5 ± 1.3	1.0 ± 0.8
Kernel	RSSI	6	1.3 ± 1.0	1.0 ± 0.7
FILA	CSI	1	4.4 ± 2.4	2.2 ± 1.4
Biloc	CSI	1	4.8 ± 2.3	2.5 ± 2.0

of a normal person [40,41]. The user location was updated every Δt =1 s. The testing experiments are repeated several times in different days and time.

3.4 Results And Discussions

3.4.1 SSP and Conventional Methods Comparison

The proposed SSP is added to memoryless probabilistic methods including Horus [23], DGD [51], Kernel method [46], BiLoc [7] and FILA [58]. As the maximum speed, v_{max} is pre-configured to be 4 m/s, the maximum distance a user can travel during consecutive measurements is $d_{max} = v_{max}\Delta t$ =4 m. Therefore, we used a Gaussian window with $\sigma = d_{max}$ in the SSP model.

Among all RSSI models, Horus [23] and DGD [51] are typical parametric approaches with the assumption of a single-peak Gaussian distributed $P(\boldsymbol{F}(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)$. The Kernel method [46] is a non-parametric model which builds the probability distribution by using the kernel smoothing function. The additional short term memory step is added to calculate $P(\boldsymbol{l}_i)$ following Subsection 3.2.2.

Among CSI models, FILA [58] exploits CSI amplitude to build a single-peak Gaussian distributed database $P(\boldsymbol{F}(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)$. A possible approach to calculate $P(\boldsymbol{l}_i)$ is to use Pearson correlation between the testing and the stored CSI fingerprints in the database. However, this approach relies only on the spatial information, which ignores the time related information from the previous location. Instead, SSP model

determines $P(\boldsymbol{l}_i)$ based on the user's previous location which adds the valuable information from time domain. On the other hand, BiLoc [7] exploits bi-modal data as the fingerprints. During training, a deep autoencoder network with a pre-train restricted Boltzmann machine (RBM) is used to extract the unique channel features. In the testing, following the radial basis function (RBF), a Bayesian probability model is employed to estimate position. The probability $P(\boldsymbol{F}(\boldsymbol{l}_{curr})|\boldsymbol{l}_i)$ is measured according to the difference between the original and reconstructed CSI. The probability $P(\boldsymbol{l}_i)$ is assumed to be uniformly distributed. In contrast, SSP model adds an additional short term memory step to BiLoc similar to RSSI cases.

Table 3.2 illustrates the average localization errors of probabilistic algorithms before and after applying SSP. The chosen parameters of SSP are Gaussian window (Eq. (3.3)) with $\sigma = d_{max}$. Clearly, with the additional sequential consideration, the proposed SSP models show the significant performance improvement comparing with the original approaches. For Horus, the average error decreases 33% from 1.5±1.2 m to as low as 1.0±0.7 m by applying SSP. Similarly, SSP boosts the performance of the DGD and Kernel method by ∼33% and ∼25% respectively. In the cases of CSI fingerprinting localization, SSP helps to improve the accuracy of FILA by ∼50%, from 4.4±2.3 m to 2.2±1.4 m. Among all models, the most significant improvement is achieved in the case of BiLoc with 48% reduction of the average error from 4.8±2.3 m to 2.5±2.0 m.

In addition to the reduction of average error, SSP also reduces the maximum error of all conventional methods significantly. Fig. 3.4 compares the cumulative distribution function (CDF) of localization errors of Horus, DGD and Kernel method before and after adopting SSP. Due to larger RSSI fluctuations, the memoryless approaches may choose a wrong location with similar fingerprints, which could be far from the actual location, leading to an extreme large error in the scale of the testing site di-

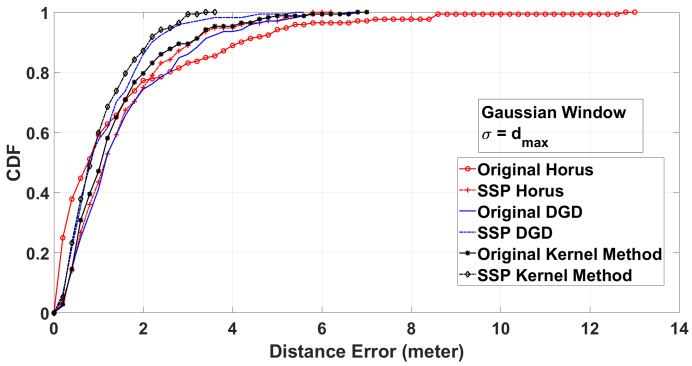


Figure 3.4: CDF of the localization error of conventional probabilistic models and SSP models with RSSI fingerprint.

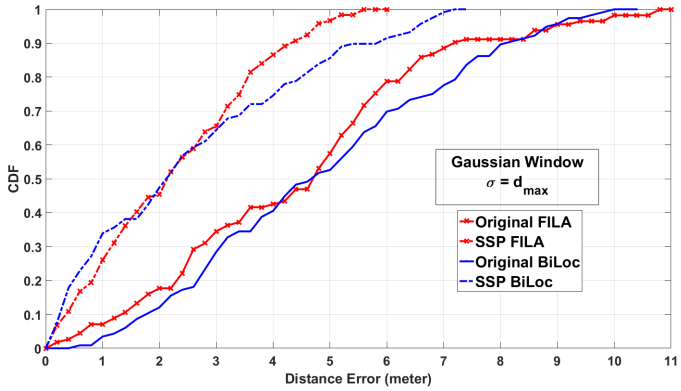


Figure 3.5: CDF of the localization error of conventional probabilistic models and SSP models with CSI fingerprint.

mension. As shown in Fig. 3.4, without SSP, the maximum error can be as high as 13 m for Horus and 7 m for DGD and Kernel method. In contrast, SSP eliminates such error pattern with the short term memory step, resulting in a much smaller maximum error of 6 m for Horus, 5.5 m for DGD and as low as 3.5 m for Kernel method. Besides, as shown in Fig. 3.5, for CSI experiment with 1 router, SSP helps to reduce the maximum error of FILA from 11 m to below 6 m and BiLoc from 10 m to $\sim$7 m.

Evidently, the above results show that SSP can significantly boost the performance of both RSSI and CSI fingerprints based probabilistic algorithms. Besides, as shown in

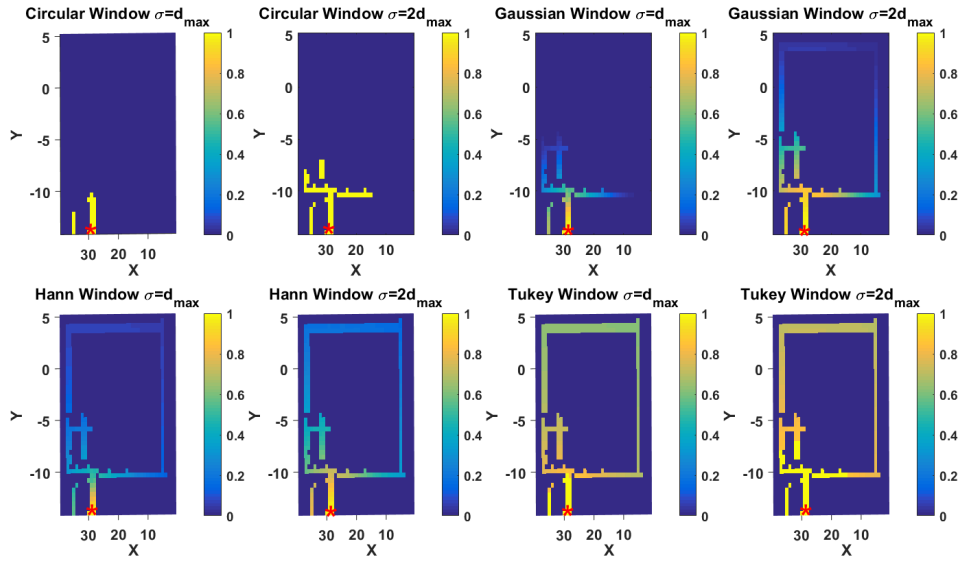


Figure 3.6: Probabilistic heat map of all RPs in the database after applying SSP windows (the red star represents for the user's previous location).

Table 3.2, the localization accuracy of the RSSI experiments with 6 APs is significantly better than that of CSI experiments with 1 AP, i.e., around 1 m of SSP models based RSSI compared with more than 2 m of SSP models based CSI. Therefore, in the following sections, to analyze important parameters such as window types and the corresponding σ value, we focus on the RSSI experiment.

3.4.2 Parameter Analysis

Window Type

Following discussions in Subsection 3.2.2, we propose three more soft range SSP windows including Gaussian, Hann and Tukey in addition to the simple hard range circular window. Fig. 3.2 illustrates the shapes of these windows. Among all, the circular window has a clear drop at the circumference $D = \sigma$, which eliminates the possibility of any RPs having the distances bigger than σ being considered as close to the current localization. Gaussian and Hann windows have the smooth decline curves

Table 3.3: Average Errors of SSP Horus(meter)

σ (m)	Circular	Gaussian	Hann	Tukey
$d_{max}/2$	1.8 ± 2.0	1.1 ± 1.0	1.0 ± 0.9	1.1 ± 0.8
d_{max}	1.2 ± 0.8	1.0 ± 0.7	0.9 ± 0.7	1.2 ± 0.9
$3d_{max}/2$	1.2 ± 0.9	1.1 ± 0.8	1.1 ± 0.8	1.2 ± 0.9
$2d_{max}$	1.3 ± 0.9	1.2 ± 0.9	1.1 ± 0.8	1.2 ± 0.9

with the increase of distance D. When σ increases, these curves spread out more. On the other hand, Tukey window is the combination between hard range and soft range forms with the equivalent highest probability for all RPs with distance smaller than σ and the gradual dropping curve following the increase of distance D.

Fig. 3.6 demonstrates the effects of various SSP windows by their probabilistic heat maps with the color scale to the probability density. The red star in the map is the predicted previous location $\boldsymbol{l}'_{prev}$. For each window we plotted both $\sigma = d_{max}$ and $\sigma = 2d_{max}$. Clearly, the circular window only takes the area around the previous location with the highest probabilities (yellow color area) and eliminates the rest (blue color area). Gaussian window yields the most significant change when we increase the value of σ. When $\sigma = d_{max}$, a half of the area is ignored, while when $\sigma = 2d_{max}$, most of the RPs are included with positive probabilities. Besides, Hann and Tukey have less dramatic changes than Gaussian when we increase σ. Furthermore, using Tukey window, the colors of different RPs are similar across the whole maps, indicating fewer differences among the probability densities of different RPs than the other window types.

Standard deviation σ

The width of the short term memory windows is determined by the standard deviation σ. Fig. 3.2 shows that when σ increases, the PDF spread out more, which means more RPs being included as candidate locations in SSP. If σ is too small, many possible RP candidates will be ignored, leading to huge accumulated errors. On the other

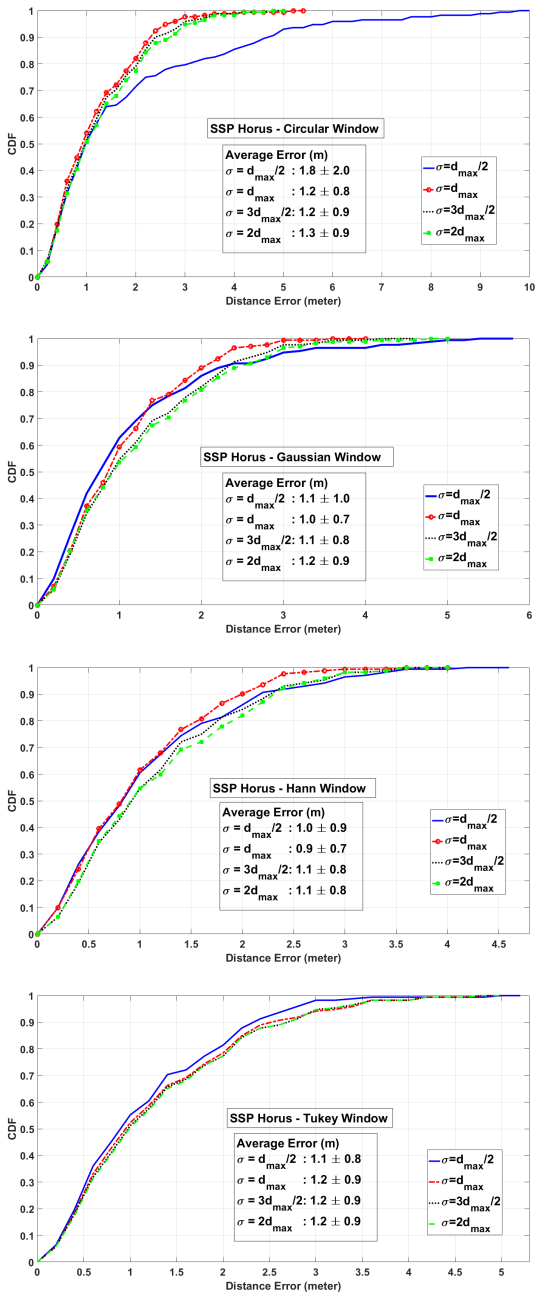


Figure 3.7: CDF of the localization error of SSP Horus model with different window forms and σ values (a) Circular window. (b) Gaussian window. (c) Hann window. (d) Tukey window.

hand, if σ is too large, ambiguous locations are introduced which severely affects the localization performance.

Table 3.4: Average Errors of SSP DGD(meter)

σ (m)	Circular	Gaussian	Hann	Tukey
$d_{max}/2$	1.8 ± 2.4	1.0 ± 0.9	1.1 ± 0.9	1.2 ± 0.9
d_{max}	1.2 ± 0.9	1.1 ± 0.8	1.1 ± 0.8	1.1 ± 0.9
$3d_{max}/2$	1.3 ± 1.0	1.1 ± 0.9	1.1 ± 0.9	1.3 ± 1.0
$2d_{max}$	1.3 ± 1.0	1.2 ± 1.0	1.2 ± 0.9	1.3 ± 1.1

Table 3.5: Average Errors of SSP Kernel Method (meter)

σ (m)	Circular	Gaussian	Hann	Tukey
$d_{max}/2$	1.2 ± 1.0	0.9 ± 0.8	0.9 ± 0.7	1.0 ± 0.8
d_{max}	1.2 ± 0.9	1.0 ± 0.7	1.0 ± 0.8	1.0 ± 0.8
$3d_{max}/2$	1.1 ± 0.8	1.0 ± 0.8	1.0 ± 0.8	1.0 ± 0.8
$2d_{max}$	1.1 ± 0.9	1.1 ± 0.8	1.1 ± 0.8	1.1 ± 0.9

In order to study the impacts of σ on SSP, Table 3.3, 3.4 and 3.5 compare the average errors with different σ when SSP model is applied. For the circular window, all of the RPs having a distance bigger than σ are not considered in the current location estimation. Therefore, a significant error is observed with a low σ value, e.g., 1.8 ± 2.0 m at $\sigma = d_{max}/2$ with SSP Horus and SSP DGD. When σ increases, the performance is improved with the average error being around 1.2 ± 0.8 m at $\sigma = d_{max}$. The lowest average error of 1.1 ± 0.8 m is reached when $\sigma = 3d_{max}/2$. In contrast, the performance of the soft range windows such as Gaussian, Hann and Tukey, is insensitive to σ value. When $\sigma = d_{max}$, soft range windows with Horus and DGD have the lowest average errors of 1.0 ± 0.7 m and 1.1 ± 0.8 m respectively. On the other hand, Kernel method reaches its best performance of 0.9 ± 0.8 m at $\sigma = d_{max}/2$. When σ increases from $\sigma = d_{max}$ to $\sigma = 2d_{max}$, there is only a slight change in the performance of all three types of soft range windows. The reason is that instead of a hard cutoff, soft range windows provide a smooth bias toward locations closer to the previous prediction while no locations are fully eliminated from consideration.

Fig. 3.7 illustrates SSP Horus error CDFs with different σ values. For the circular, Gaussian and Hann windows, when $\sigma = d_{max}/2$ the maximum errors of SSP Horus increase significantly to 10 m, 5.7 m and 4.5 m respectively. When σ equals or is larger

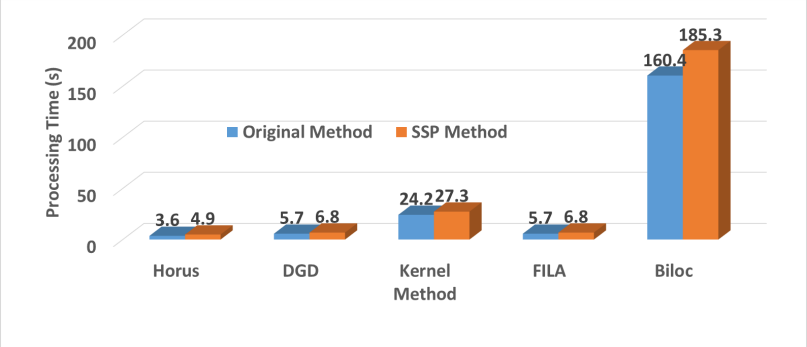


Figure 3.8: Processing time of SSP methods compared with the conventional methods.

than d_{max}, all maximum errors are around 4-5 m. Especially, for Tukey window, the CDFs are similar even when σ changes from $d_{max}/2$ to $2d_{max}$.

In summary, $\sigma = d_{max}$ provides the best localization accuracy for most of the cases. Furthermore, Gaussian and Hann windows consistently show the comparable performance. Therefore, in the rest of the analysis, we choose Gaussian window at $\sigma = d_{max}$ as our SSP window.

3.4.3 Performance Analysis

The conventional probabilistic approaches using Eq. (3.2) have a complexity $O(N)$, where N is the number of features as explained in Subsection 3.2.1. SSP adds a simple short term memory according to Eq. (3.3), (3.4) and (3.5) with no significant addition to the overall complexity. Fig. 3.8 shows the comparison of the processing time among all methods with and without SSP. The computations are estimated on a Intel core i5-3320M 2.6 GHz based computer and an NVidia Geforce FTX 1050 GPU is used for training and testing neural network. For a fair comparison, the shown processing time here is for the testing phase. The processing time of SSP for all five implemented methods, including Horus, DGD, Kernel method, FILA and Biloc, is slightly higher than the original ones, ranging from 10% to 20% difference. In summary, the processing time of SSP model is comparable to that of the other

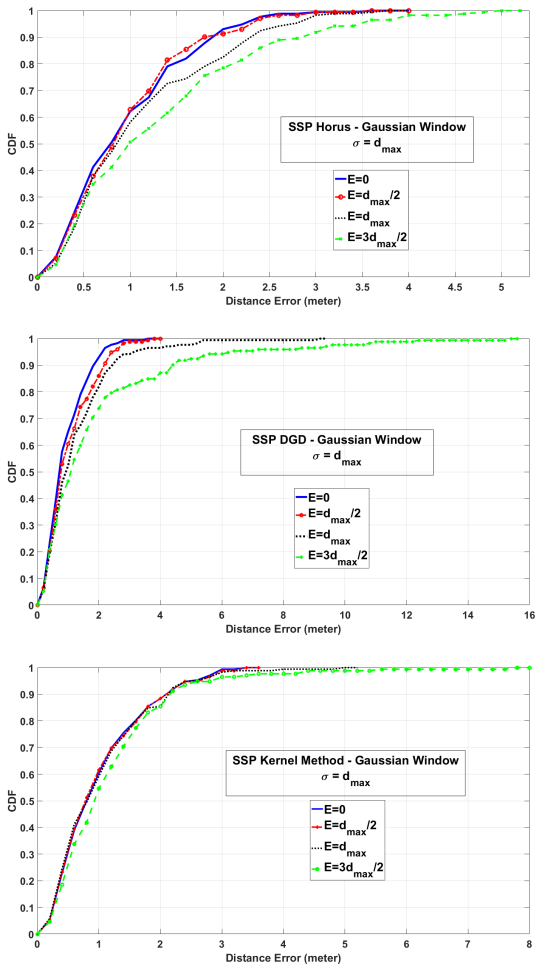


Figure 3.9: CDF of localization errors of SSP based models using RSSI fingerprint in different error scenarios of historical data. (a) Horus. (b) DGD. (c) Kernel method.

conventional probabilistic approaches.

Since SSP leverages the information of a user's previous position to estimate the current location, the performance of SSP depends on the accuracy of historical data. Note that all of our SSP results are based on the estimated history location with non-zero errors. In order to estimate the propagation error due to the imperfect prior location estimation, Fig. 3.9 illustrates the localization errors of SSP Kernel with both the ideal and erroneous history data. Starting with the perfect historical coordinate $\boldsymbol{h}(x, y)$ for every location in the testing trajectories mentioned in Section 3.3, an

amount of error E is added to $\boldsymbol{h}$. The erroneous prior location $\boldsymbol{h'}(x', y')$ is obtained as: $x' = x + x_e$, $y' = y + y_e$, where x_e and y_e are random variables that follow Gaussian distribution

$$x_e \sim \mathcal{N}(0, \sigma_{x_e}^2)\,;\; y_e \sim \mathcal{N}(0, \sigma_{y_e}^2)\,;\; \sqrt{\sigma_{x_e}^2 + \sigma_{y_e}^2} = E$$

Fig. 3.9 shows the cases where E is proportional to $d_{max} = 4$ m. Clearly, if the error E of the historical data is within $d_{max}/2$, the localization accuracy of all three methods are similar to the ideal case with a maximum error of $\sim$4 m. When E increases to d_{max}, the accuracy of SSP Horus becomes slightly worse with 80% of the error is under 2 m compared with 1.5 m of the ideal cases. On the other hand, SSP DGD and SSP Kernel are affected more significantly with the maximum error increasing to 9 m and 5 m respectively. As shown in Table 3.3, 3.4 and 3.5, average errors of all SSP models are around $d_{max}/4 \approx 1$ m, which indicates that SSP is robust to localization error of the previous position. If the value of error E is larger than d_{max}, i.e., $E = 3d_{max}/2$, the performance will degrade and the accumulated errors start increasing. In order to solve the problem, the stationary time in a regular walking pattern can be utilized. According to the survey in Ref. [47], the percentage of stationary time exceeds 80% for most mobile users. During the no movement period, the number of RSSI readings collected in one-location (S_2) is sufficient to improve the conventional probabilistic model accuracy. Therefore, in order to enhance the accuracy when locating a user's position in a long trajectory, we can employ these stationary locations as alignment points where the prior locations can be ignored. In that case, the conventional probabilistic approaches can be exploited to estimate the user's location.

3.5 Conclusions

In conclusion, we have proposed a simple but efficient semi-sequential probabilistic model, which applies an additional short term memory step to enhance the performances of the indoor localization probabilistic approaches. This model leverages the information of the previous position to effectively determine the candidate location probability since the user's speed in an indoor environment is bounded. Three soft range windows including Gaussian, Hann and Tukey have been proposed as SSP models. Several experiments have been conducted including both RSSI and CSI fingerprints to demonstrate that SSP reduces the maximum error and boosts the performance of other existing probabilistic approaches by at least $25\% - 30\%$. The error analysis also shows that the proposed SSP model is robust to localization error of the previous position.

Chapter 4

Recurrent Neural Networks For Accurate RSSI Indoor Localization

4.1 Introduction

As mentioned in the previous chapters, among WiFi RSSI fingerprinting indoor localization approaches, artificial neural network (ANN) [5, 19] estimates location nonlinearly from the input by a chosen activation function and adjustable weightings. In indoor environments, because the transformation between the RSSI values and the user's locations is nonlinear, it is difficult to formulate a closed form solution [3]. ANN is a suitable and reliable solution for its ability to approximate high dimension and highly nonlinear models [5]. Recently, several ANN localization solutions, such as multilayer perceptron (MLP) [62], robust extreme learning machine (RELM) [63], multi-layer neural network (MLNN) [64], convolutional neural network (CNN) [65], etc., have been proposed.

Although having been extensively investigated in the literature, all of the existing indoor localization algorithms still face challenges such as spatial ambiguity, RSSI in-

stability and RSSI short collecting time per location [8]. To address these challenges, this chapter focuses on recurrent neural network (RNN) to determine the user's location by exploiting the sequential correlation of RSSI measurements. Since the moving speed of the user in an indoor environment is bounded, the temporal information can be used to distinguish the locations that has similar fingerprints. Note that resolving the ambiguous locations has been a common challenge in indoor localization. Some works in literature also exploit the measurements in previous time steps to locate the current location, including the use of Kalman filter [24–27] and soft range limited K-nearest neighbors (SRL-KNN) [8]. Among them, Kalman filter estimates the most likely current location based on prior measurements, assuming a Gaussian noise of the RSSI and linear motion of the detecting object. However, in real scenarios, these assumptions are not necessarily valid [28]. In comparison, SRL-KNN does not make the above assumptions but requires that the speed of the targeting object is bounded, e.g., from 0.4 m/s to 2 m/s. If the speed of the target is beyond the limit, the localization accuracy of SRL-KNN will be severely impaired. In contrast, our RNN model is trained from a large number of randomly generated trajectories representing the natural random walking behaviours of humans. Therefore, it does not have the assumptions or constraints mentioned above.

The main contributions of this chapter are summarized as follows.

1 According to our knowledge, there is no existing comprehensive RNN solution for WiFi RSSI fingerprinting with detailed analysis and comparisons. Therefore, we propose a complete study of RNN architectures including network structures and parameter analysis of several types of RNNs, such as vanilla RNN, long short term memory (LSTM) [10], gated recurrent unit (GRU) [11], bidirectional RNN (BiRNN), bidirectional LSTM (BiLSTM) [12] and bidirectional GRU (BiGRU) [13].

Table 4.1: Comparisons of Indoor Localization Experiments Using Machine Learning Techniques

Method	Feature	Access point (AP)	Reference point (RP)	Testing Point	Grid Size	Accuracy
MLP [62]	RSSI	6	207	50	1.7 m	2.8 ± 0.1 m
DANN [19]	RSSI	15	45	46	2 m	2.2 ± 2.0 m
RELM [63]	RSSI	8	30	10	3.5 m	3.7 ± 3.4 m
MLNN [64]	RSSI	9	20	20	1.5 m	1.1 ± 1.2 m
ConFi [5]	CSI	1	64	10	1.5 - 2 m	1.3 ± 0.9 m
Geomagnetic RNN [66]	Magnetic Information	-	629	5% of RPs	0.57 m	1.1 m

2 The proposed models are tested in two different datasets including an in-house measurement dataset and the published dataset UJIIndoorLoc [48]. The accuracy is compared not only with the other neural network methods, i.e., MLP [62] and MLNN [64], but also some popular conventional methods, i.e., RADAR [21], SRL-KNN [8], Kernel method [46] and Kalman filter [24].

3 Three challenges of WiFi indoor localization, i.e., spatial ambiguity, RSSI instability and RSSI short collecting time per location are discussed and addressed. Furthermore, the other important factors, including the network training time requirement, users speed variation, different testing time slots and historical prediction errors, are discussed and analyzed.

4.2 Related Works

The first research on neural network for indoor localization was reported by Battiti *et al.* [18, 62]. In that work, a multilayer perceptron network (MLP) with 3 layers consisting one input layer, one hidden layer with 16 neurons and one output layer was implemented to nonlinearly map the output (coordinate) from the input (RSSI). There are 207 RPs for training and 50 random points for validation. The accuracy of this model is 2.82±0.11 m, which is comparable with that of simple K nearest neighbours (KNN) algorithm [18]. In order to achieve better performance, multi-layer feed-forward neural network (MLNN) [64] with 3 hidden layers in their

experiment. MLNN is designed with 3 sections: transforming RSSI section, denoising RSSI section and locating section with the boosting method to tune the network parameter if the misclassified data is detected. The other refinement of MLP is proposed in discriminant-adaptive neural network (DANN) [19], which inserts discriminative components (DCs) layer to extract useful information from the inputs. The experiment shows that DANN improves the probability of the localization error below 2.5 m by 17% over the conventional KNN RADAR [21]. As all of those above methods are time-consuming in training phase, robust extreme learning machine (RELM) [63] is proposed to increase the training speed of the feed-forward neuron networks. RELM consists of a generalized single-hidden layer with random hidden nodes initialization and kernelized formulations with second order cone programming. The experiment illustrates that the training speed of RELM is over 100 times faster than the conventional machine learning methods [67] and the localization accuracy is increased by 40%.

Although the feed-forward neural networks are simple and easy to implement, they can not extract the useful information efficiently from the noisy WiFi signal, leading to a limited accuracy. Therefore, more complicated neural networks were adopted for indoor WiFi localization, e.g., convolutional neural network (CNN) and recurrent neural network (RNN). ConFi [5] proposed a three layers CNN to extract the information from WiFi channel state information (CSI). CSI from different subcarriers at different time is arranged into a matrix, which is similar to an image. ConFi is trained using these CSI feature images collected at a number of RPs. The localization results is the weighted centroid of RPs with high output value. The experiment illustrates that ConFi outperforms conventional KNN RADAR [21] by 66.9% in term of mean localization error. However, as mentioned in Section 4.1, it is not easy to get CSI because only some specific wireless network interface cards, e.g, Intel

WiFi Link 5300 MIMO NIC, Atheros AR9390 or Atheros AR9580 chipset can obtain that information. Recently, RNN is used to locate a user's position in an indoor environment. [68] proposed a simple RNN with 1 time step and 2 hidden layers. RNN classifies 42 RPs based on the RSSI readings from 177 APs. The classification accuracy is 82.47%. A more efficient RNN is published in [66]. In this work, the RNN has 200 neurons uses mean squared error (MSE) loss function and has 20 time steps trace as the input for the network. In that work, geomagnetic data is used as fingerprints instead of the wireless data. A million traces of various pedestrian walking patterns are generated with 95% of them for training and 5% for validation. The achieved localization errors range from 0.441 to 3.874 m with the average error being 1.062 m. In addition to the geomagnetic data, the light intensity is also utilized in the deep LSTM network [69] to estimate the indoor location of the target mobile device. Their 2 layers LSTM exploits temporal information from bimodal fingerprints, i.e., magnetic field and light intensity data, through recursively mapping the input sequence to the label of output locations. The accuracy is reported as 82% of the test locations with location errors around 2 m and the maximum error being around 3.7 m.

Table 4.1 summarizes the experimental set-up and the results from the above mentioned neural network methods. Here the number of access points (AP), reference points (RP) and testing points vary between those experiments and the grid size is defined as the distance between two consecutive RPs.

In general, those above methods provide acceptable accuracy from 1 m to 3 m but none of them sufficiently investigated the three problems of using RSSI fingerprints as stated in Section 4.1, i.e., spatial ambiguity, RSSI instability, RSSI short collecting time per location. Furthermore, according to the authors' knowledge, there is no comprehensive RNN solution using WiFi RSSI fingerprinting with appropriate

analysis, details and comparisons. Therefore, we propose a detailed RNN solution for fingerprint indoor localization using WiFi to solve the above three challenges. Besides, different types of RNN including vanilla RNN, long short-term memory, gated recurrent unit, and bidirectional LSTM and their structure with all of the important parameters will be analysed in detail. Our localization results of RNN are compared not only with the other neural network methods, i.e., MLP [62], MLNN [64] but also some conventional methods including KNN [21], Kernel method [46] and Kalman filter [24] using 2 different databases, one in house measurements and one publicly available data source.

4.3 RNN Methods

4.3.1 Recurrent Neural Network Overview

A recurrent neural network is a class of artificial neural network where the output results depend not only on the current input value but also on the historical data [66]. RNN is often used in situations where data has a sequential correlation. In the case of indoor localization, the current location of the user is correlated to its previous locations as the user can only move along a continuous trajectory. Therefore, RNN is appropriate to exploit the sequential RSSI measurements and the trajectory information to enhance the accuracy of the localization. Up to date, there are several RNN variants because the simplest one, vanilla RNN [70] has limited applications due to vanishing gradient during the training phase [71]. To mitigate that effect, long short term memory [10] creates an internal memory state which adds the forget gate to control the time dependence and effects of the previous inputs. GRU [11] shares a similar idea to that of LSTM but reduces from 3 gates of LSTM, i.e., forget, update and output gate, to only 2 gates, i.e., update and reset gate. BiRNN [70],

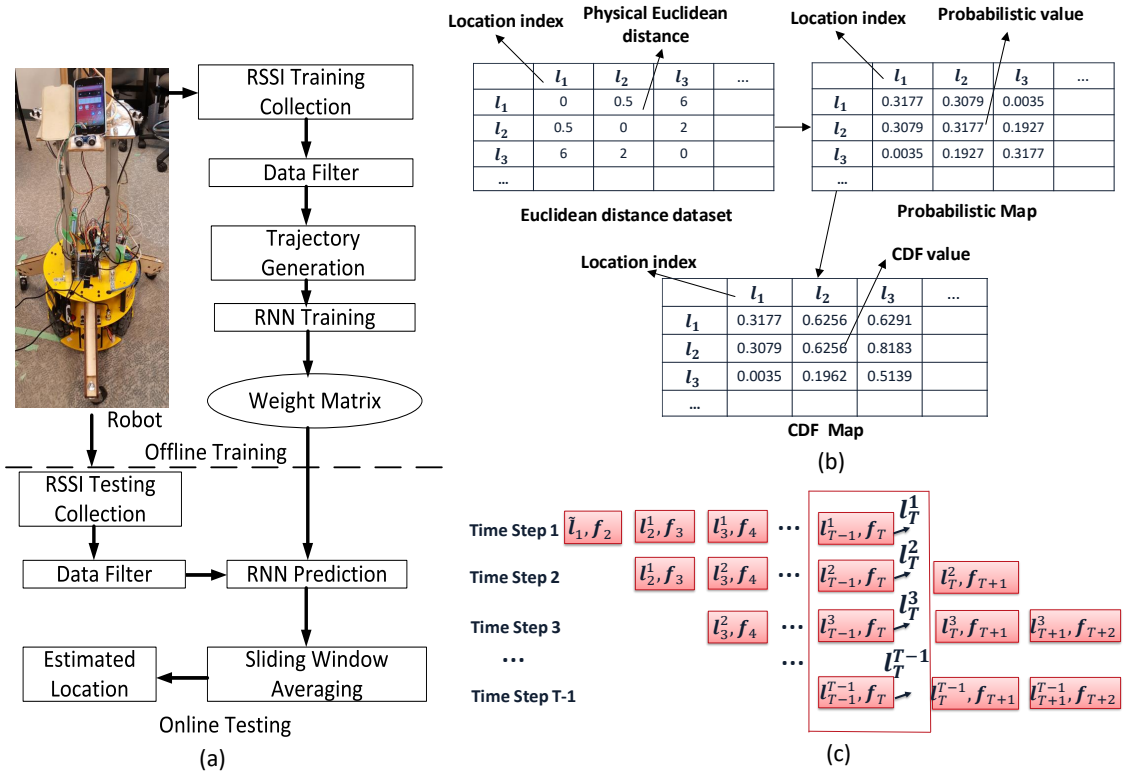


Figure 4.1: (a) Localization process of the proposed RNN system. (b) Trajectory generation process. (c) Sliding window averaging in online testing phase.

BiLSTM [12] and BiGRU [13] are the extensions of the traditional RNN, LSTM and GRU respectively, which not only utilizes all available input information from the past but also from the future of a specific time frame. In the following section we will describe details about our proposed WiFi RSSI indoor localization system using RNN model.

4.3.2 Proposed Localization System

The fingerprinting localization system is generally divided into two phases: a training phase (offline phase) and a testing phase (online phase). In the training phase, RSSI readings at each predefined reference point (RP) location, are collected and stored to a database as fingerprints. Assuming that the area of interest has P APs and M

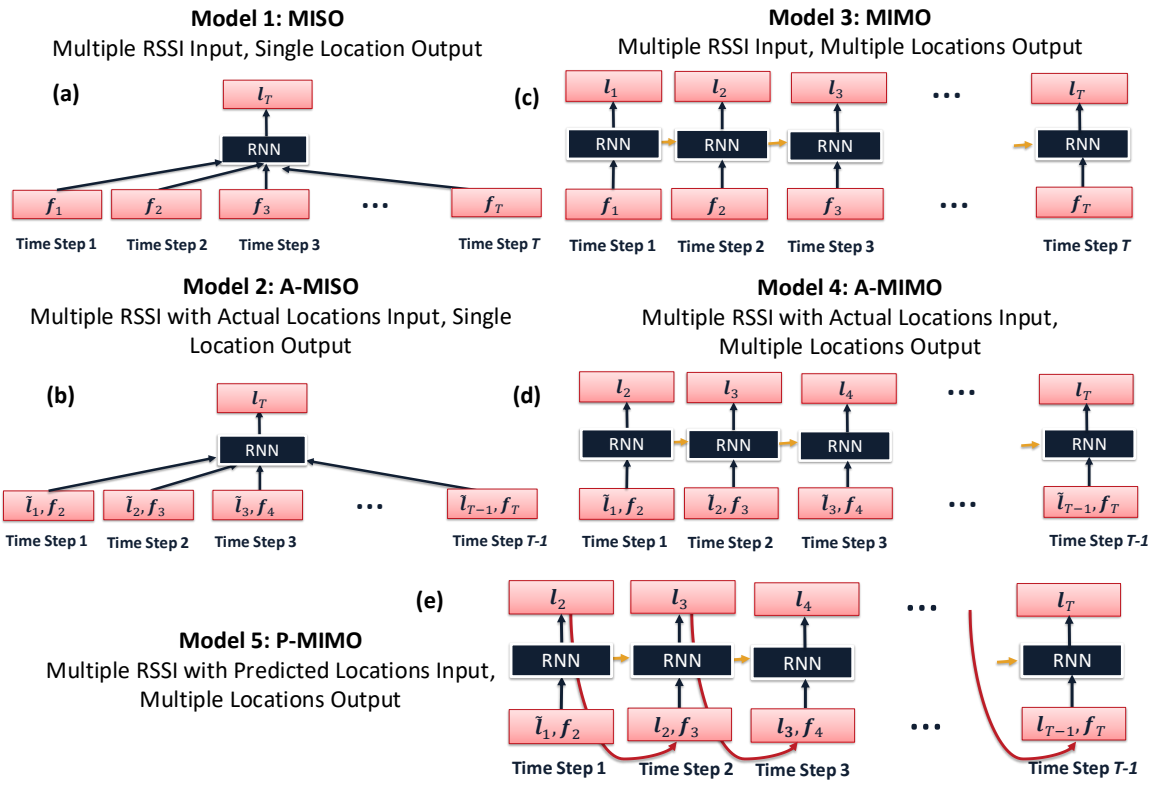


Figure 4.2: Proposed RNN models.

RPs, we consider each RP i at its physical location $\boldsymbol{l}_i(x_i, y_i)$ having a corresponding fingerprint vector $\boldsymbol{f}_i = \{F_1^i, F_2^i, ..., F_N^i\}$, where N is the number of available features and $F_j^i (1 \leq j \leq N)$ is the j-th feature at RP i. In the testing phase, each unknown location of the user, denoted as a testing point, is determined by the localization algorithm. During the training phase, a set of RSSI readings (S_1 scans) are collected at a single RP while in the testing phase, and only a small number of RSSI readings (S_2 scans), e.g., $S_2 = 1$ or $S_2 = 2$, is available as the user is expected to frequently moving in practical scenario. Fig. 2.1(a) illustrates the localization map with 6 APs, 365 RPs and 175 testing points, while Fig. 2.1(b) shows the heat map of these 6 APs. The architecture of the proposed RNN system is presented in Fig. 4.1(a). The RSSI data for both training and testing will be collected with the support of an autonomous

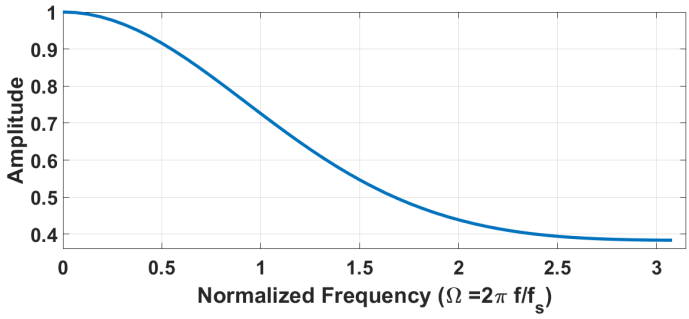


Figure 4.3: Weighted average filter transfer function.

robot. Details of the process will be described as follows.

Data Filter

The RSSI collected by a client device often experiences substantial fluctuations due to dynamically changing environments such as human blocking and movements, interference from other equipment and devices, receiver antenna orientation, etc., [28, 42]. In our experiment, the mobile device was put on the location $P(7, 4)$ as shown in Fig. 2.1(a). The experiment was conducted in busy hours when many students (up to 10) used WiFi and walked around the lab. A maximum RSSI standard deviation of 5.5 dB was recorded over 100 consecutive RSSI readings, and 5% of the measurements (5 readings) could not be detected (missing values). In those cases, the client device missed the beacon frame packets sent by the router. In order to filter out those outliers and make the measurement valid, we adopt the iterative recursive weighted average filter [72]. Based on the transfer function in Fig. 4.3, the weighted average filter has the form of a low pass filter. The effectiveness of the filter will be studied in Section 4.5.

Trajectory Generation

The RSSIs at the output of the data filter will be used to generate random training trajectories under the constraints that the distance between consecutive locations is bounded by the maximum distance a user can travel within the sample interval in practical scenarios. Fig. 4.1(b) illustrates the trajectory generation process. Firstly, the physical Euclidean distance between each location and the rest of the database is calculated to form a Euclidean distance dataset. Secondly, based on that, the probabilistic map will be generated to represent the probability of a location ($P(\boldsymbol{l}_i)$) that will become the next location of the user in the trajectory. Since the moving speed of an indoor user is limited, the locations which are near to the previous locations should have higher probability to become the next location in the trajectory than the further locations. The user location will be updated in every consecutive sampling time interval Δt. Therefore, the maximum distance which the user can move in Δt is $\sigma = v_{max} \times \Delta t$, where the maximum speed v_{max} is chosen to be larger than the human indoor normal speed (from 0.4 m/s to 2 m/s [40, 41]). The normalized probability $P(\boldsymbol{l}_i)$ is calculated as follows.

$$P(\boldsymbol{l}_i) = \frac{1}{2\sigma^2(1 - e^{\frac{d_{max}^2}{2\sigma^2}})} \exp(-\frac{(x_i - x_{pre})^2 + (y_i - y_{pre})^2}{2\sigma^2}) \tag{4.1}$$

where (x_{pre}, y_{pre}) is the most recent location of $\boldsymbol{l}_i$, d_{max} is the maximum distance between the considered location $\boldsymbol{l}_i$ and the furthest location in the interested area. Eq. (4.1) has the form of a Gaussian distribution with the mean being the previous location and the standard deviation being σ. All of the locations having the same physical distance with $\boldsymbol{l}_i$ will get the same probability to be chosen as the next point on the trajectory. From the probabilistic map, a cumulative distribution function (CDF) map is built by summing the $P(\boldsymbol{l}_i)$ for each location $\boldsymbol{l}_i$. Finally, in order to

get the next location $\boldsymbol{l}_j$ of a any location $\boldsymbol{l}_i$ in the trajectory, a random number R $(0 < R < 1)$ is picked. $\boldsymbol{l}_j$ is the location which has the value in the CDF map being closest with R.

Proposed RNN Models

The proposed RNN architecture is trained by the data from consecutive locations in a trajectory to exploit the time correlation between them. Each location in a trajectory appears at a different time step. The length of a trajectory, or equivalently the number of time steps, defines the memory length T as illustrated in Fig. 4.2. The number of time steps T significantly impacts the performance of RNN because all of the weights and hidden states will be saved at every time step during a training trajectory [70]. A larger T value will incorporate more information from the past but will accumulate more localization errors. The optimal T value will be chosen by the experiment in Section 3.4.

Fig. 4.2 illustrates 5 different proposed RNN models labeled from model 1 to model 5. Model 1 has a multiple input single output (MISO) structure where several RSSI readings ($\boldsymbol{f}_i$) from the previous time steps are fed into the network to get the single location at time step T ($\boldsymbol{l}_T$). Model 2, similar to model 1, has a multiple input single output (A-MISO) structure. However, model 2 takes in actual previous step locations as well as RSSIs unlike model 1. The actual locations are the ground truth locations $\tilde{\boldsymbol{l}}$.

By comparison, model 3 to model 5 are multiple input multiple output (MIMO) structures where several RSSI readings from multiple time steps are fed to the network to get multiple output locations $\boldsymbol{l}$. Model 3 has a MIMO structure, where multiple RSSI inputs $\boldsymbol{f}_i$ produce multiple output locations $\boldsymbol{l}_i$. In contrast, model 4 takes in multiple RSSIs with multiple actual locations for the input and produces multiple

locations for the output, denoted as A-MIMO.

Model 5 takes in multiple RSSIs and multiple previous predicted locations for the input and produces multiple locations for the output, denoted as P-MIMO. Specifically, in model 4: A-MIMO, the location used in the training is the ground truth from the dataset. In model 5: P-MIMO, the location value is the predicted location ($\boldsymbol{l}_i$) from the previous time step. Note that in the testing phase, both model 2 and model 4 use the predicted locations from the previous time steps as input because ground truth is not available.

The objective of RNN training is to minimize the loss function $\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}})$ defined as the Euclidean distance between the output $\boldsymbol{l}$ and the target $\tilde{\boldsymbol{l}}$ using the backpropagation algorithm [70]. In single output models such as MISO and A-MISO, the loss function is given by

$$\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}}) = ||\boldsymbol{l}_T - \tilde{\boldsymbol{l}}_T||_2. \tag{4.2}$$

In contrast, the multiple output models MIMO, A-MIMO and P-MIMO adopt a loss function expressed by

$$\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}}) = \frac{\sum_{i=1}^{T} ||\boldsymbol{l}_i - \tilde{\boldsymbol{l}}_i||_2}{T}. \tag{4.3}$$

Sliding Window Averaging

MIMO, A-MIMO and P-MIMO have the output locations appearing in every time step (Subsection 5.2.3). In the online testing phase, the output location $\boldsymbol{l}_T$ will appear in several time steps as shown in Fig. 4.1(c). $\boldsymbol{l}_i^j$ is the output location $\boldsymbol{l}_i$ of time step j. At the output time step $T-1$, we have a set of output location $(\boldsymbol{l}_T^1, \boldsymbol{l}_T^2, ..., \boldsymbol{l}_T^{T-1})$ from $T-1$ previous steps. In each output time step, the accuracy of the targeted output result $\boldsymbol{l}_T$ can be slightly different due to the difference in length of previous historical information. For example, in Fig. 4.1(c), at output time step 1, $\boldsymbol{l}_T^1$ is estimated with the information of $T-1$ previous steps. However, at output time step 2, the number

of previous steps for $\boldsymbol{l}_T^2$ decrease to $T-2$; and at time step $T-1$, $\boldsymbol{l}_T^{T-1}$ is predicted with no previous step. Therefore, the sliding window averaging can average the error of the predicted location $\boldsymbol{l}_T$ in multiple output steps and increase the localization accuracy. The final output result $\boldsymbol{l}_T$ will be the average of the above output set:

$$\boldsymbol{l}_T = \frac{\sum_{j=1}^{T-1} \boldsymbol{l}_T^j}{T-1}. \tag{4.4}$$

4.4 Database And Experiments

All experiments have been carried out on the third floor of Engineering Office Wing (EOW), University of Victoria, BC, Canada. The dimension of the area is 21 m by 16 m. It has three long corridors as shown in Fig. 2.1(a). There are 6 APs and 5 of them provide 2 distinct MAC address for 2.4 GHz and 5 GHz communications channels respectively, except for one that only operates on 2.4 GHz frequency. Equivalently, in every scan, 11 RSSI readings from those 6 APs can be collected.

The RSSI data for both training and testing will be collected using an autonomous driving robot. The 3-wheel robot as shown in Fig. 4.1(a) has multiple sensors including a wheel odometer, an inertial measurement unit (IMU), a LIDAR, sonar sensors and a color and depth (RGB-D) camera. It can navigate to a target location within an accuracy of 0.07±0.02 m. The robot also carries a mobile device (Google Nexus 4 running Android 4.4) to collect WiFi fingerprints. The dataset for offline training was collected by the phone-carrying robot at 365 RPs. At each location, 100 scans of RSSI measurements ($S_1 = 100$) were collected. To build the dataset, at each location in the training trajectory, we randomly choose one out of 100 stored RSSIs as the RSSI associated with the location. There are total 365,000 random generated training trajectories following the proposed method in Subsection 4.3.2. This approximates well the user's random walk property and helps to reduce the spatial ambiguity. The

Table 4.2: Initial setup parameters for RNN system

Category	**Value**
K-fold tests	10
RNN type	LSTM
Memory length (T)	10
Model	P-MIMO
Loss function	RMSE
Hidden layer (HL)	2
Number of neurons for each HL	100
Dropout	0.2
Optimizer	Adam
Learning rate	0.001
Number of training trajectory for 1-fold test	10,000
Number of training epochs	1000
σ	2 m
Δ_t	1 s
d_{max}	2 m

Table 4.3: Average localization errors

Method	**MISO LSTM**	**A-MISO LSTM**	**MIMO LSTM**	**A-MIMO LSTM**	**P-MIMO LSTM**
Average Error (m)	1.05 ± 0.78	0.92 ± 0.75	0.80 ± 0.67	0.91 ± 0.75	0.75 ± 0.64
Method	**RADAR [21]**	**MLNN [64]**	**MLP [62]**	**Kernel Method [46]**	**Kalman Filter [24]**
Average Error (m)	1.13 ± 0.86	1.65 ± 1.20	1.72 ± 1.17	1.10 ± 0.84	1.47 ± 1.2

initial position of the user in the whole testing trajectory is known.

In the online phase, the robot moved along a pre-defined route (Fig. 2.1(a)) with an average speed around 0.6 m/s. The robot will collect RSSI at 175 testing locations along the trajectory. At each location, only 1 or 2 RSSI scans ($S_2 = 1$ or 2) will be collected and transmitted to a server in real time. The server will predict the user's position according to the proposed algorithms and the prediction accuracy will be calculated.

4.5 Results And Discussions

The initial setup for the proposed RNN system follows the parameters presented in Table 4.2. All of the results below are presented after 10-fold tests with the total of 365,000 random training trajectories.

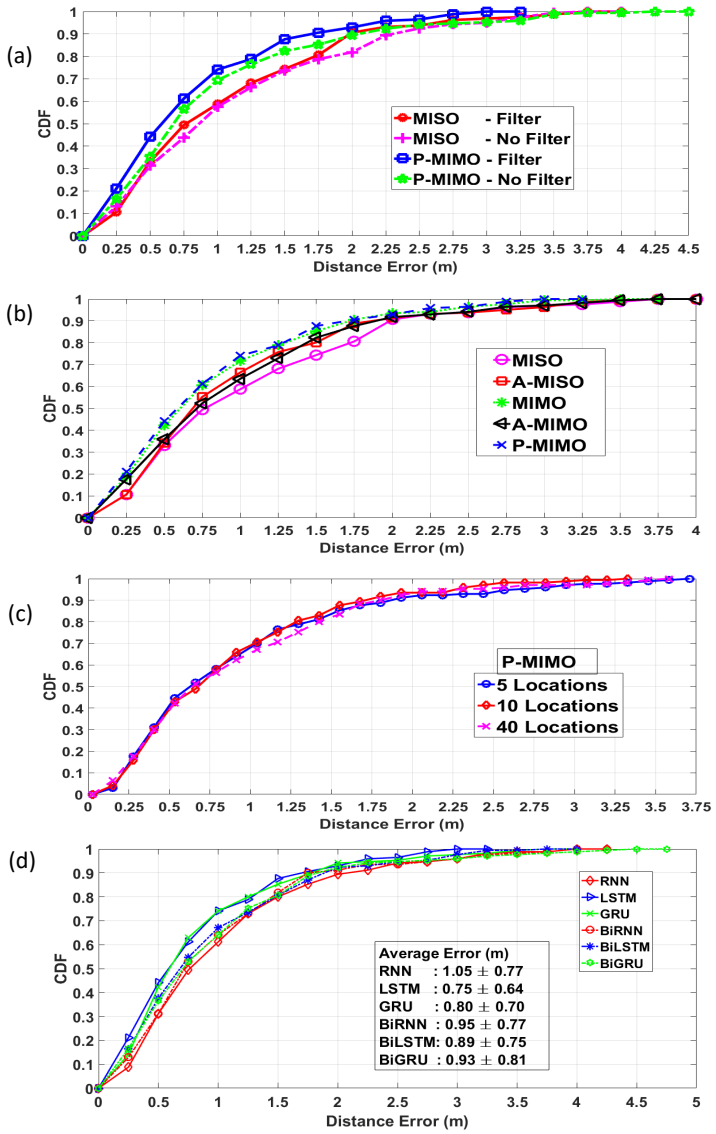


Figure 4.4: The CDF of the localization error of (a) Filter and no filter cases (b) 5 different RNN models (c) Different memory length in RNN structure (d) RNN, LSTM, GRU, BiRNN, BiLSTM and BiGRU with P-MIMO model.

4.5.1 Filter Comparison

Fig. 4.4(a) compares the cumulative distribution function (CDF) of localization errors between the proposed RNN models (i.e., MISO LSTM and P-MIMO LSTM) with 2 different cases: using or not using weighted average filter (Subsection 5.2.3) in both the dataset and testing data. For model P-MIMO (blue line), the filter helps to

decrease the maximum error from 4.5 m to 3.25 m. Besides, 80% of the error is within 1.25 m with the filter while no filter case is 1.5 m. In model MISO, the filter also illustrates better performance with 80% of the error being within 1.75 m compared with 2 m of the no filter case. In the tests of A-MISO, MIMO and A-MIMO, the weighted average filter also provides consistently good results. The localization accuracy of those methods has approximately 15% improvement with the use of filter, e.g., 0.80±0.60 m for A-MISO with the filter compared with 0.95±0.75 m for A-MISO without the filter. Because of the striking improvements, we adopted the filter to all models proposed in the article.

4.5.2 Model Comparison

Table 4.3 illustrates the average errors of all proposed models in Subsection 5.2.3. Among them, P-MIMO achieves the best performance with an accuracy of 0.75±0.64 m. MIMO is the second best performer with the average error of 0.80±0.67 m. MISO has the worst accuracy with the error of 1.05±0.78 m. Fig. 4.4(a) compares the CDF errors among these five models. MIMO and P-MIMO consistently show the dominating accuracy with 80% of the errors within 1.2 m, compared with 1.7 m of MISO. Furthermore, the maximum error of P-MIMO is 3.25 m which is lower than the 4 m obtained from MISO. In the rest of the chapter, P-MIMO is chosen for further performance study.

4.5.3 Hyper-parameter Analysis

This subsection provides a detailed study of choosing the optimal hyper-parameters for the proposed model P-MIMO. Although the 5 proposed models, i.e., MISO, A-MISO, MIMO, A-MIMO, P-MIMO, have different input and output, their general neural network structures are mostly similar. Therefore, the procedure to select the

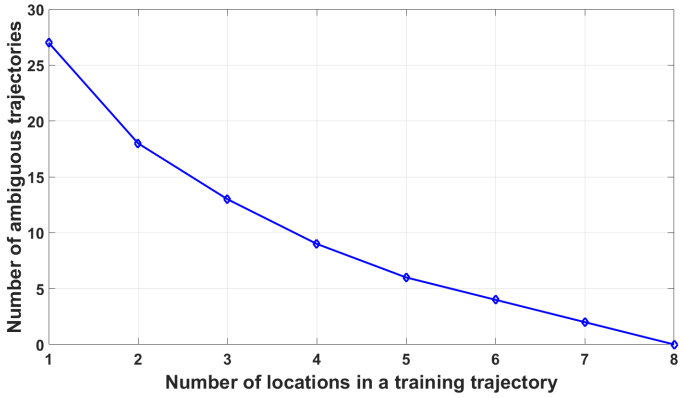


Figure 4.5: Average number of ambiguous trajectories with different number of locations in a training trajectory

optimal hyper-parameters of P-MIMO will also be valid for the other proposed models including MISO, A-MISO, MIMO and A-MIMO.

Memory Length T

Fig. 4.4(c) shows the results of P-MIMO LSTM with different memory length (Subsection 5.2.3), i.e., a training trajectory has 5, 10 or 40 locations. The performance of all three cases is comparable. The 10 time steps training trajectory has a slightly better accuracy with the maximum error being only 2.9 m, compared with 3.5 m and 3.75 m in the case of 40 time steps and 5 time steps respectively.

The theoretical explanation is as follows. A location $\boldsymbol{l}_j$ is defined as an ambiguous point of $\boldsymbol{l}_i$ if their physical distance is larger than the grid size but their two vectors $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ have high Pearson correlation coefficient above the correlation threshold. Besides, two locations are defined as physical neighbours if the physical distance between them is less than or equal to the grid size. The correlation threshold is chosen based on the average correlation coefficients between $\boldsymbol{l}_i$ and all of its physical nearest neighbours, i.e., approximately 0.9 in our database. Then all non-nearest-neighbour locations whose correlation coefficient above this threshold are considered

as ambiguous points. Pearson correlation coefficient $\rho(\boldsymbol{f}_i, \boldsymbol{f}_j)$ between $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ can be calculated as follows

$$\rho(\boldsymbol{f}_i, \boldsymbol{f}_j) = \frac{1}{N-1}\sum_{n=1}^{N}(\frac{F_n^i - \mu_i}{\delta_i})(\frac{F_n^j - \mu_j}{\delta_j}) \tag{4.5}$$

where μ_i, μ_j are the mean of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ respectively, δ_i, δ_j are the standard deviation of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ respectively. Similar to the definition of the ambiguous location, 2 trajectories are defined as ambiguous if they include different locations but the combinations of their fingerprints have a high Pearson correlation coefficient. Fig. 4.5 demonstrates the advantage of the proposed LSTM model which exploits the sequential trajectory locations compared with the conventional single point prediction. In the case of single point prediction (1 location in a training trajectory), the average number of ambiguous locations in our database are 27. If we increase the number of locations in a trajectory for training, the number of ambiguous trajectories decrease significantly. If a training trajectory has more than 8 locations, there will be no ambiguity in our database. Therefore, our memory length configured before as 10 locations is reasonable to remove all of the ambiguity.

Number of Hidden Layers and Neurons

Fig. 4.6(a) illustrates the average localization errors of P-MIMO LSTM with different number of hidden layers and neurons per layer. In general, the model with 2 hidden layers has better accuracy than the model of 1 hidden layer. Regarding the 1 hidden layer model, 200 neurons lead to the best accuracy of 0.85±0.72 m. Increasing more neurons does not result in performance gain. By comparison, the best accuracy of 2 hidden layers model is 0.75±0.64 m when the number of neurons per layer is 100. In summary, 2 hidden layers and 100 neurons per layer are the optimal parameters for our proposed LSTM network.

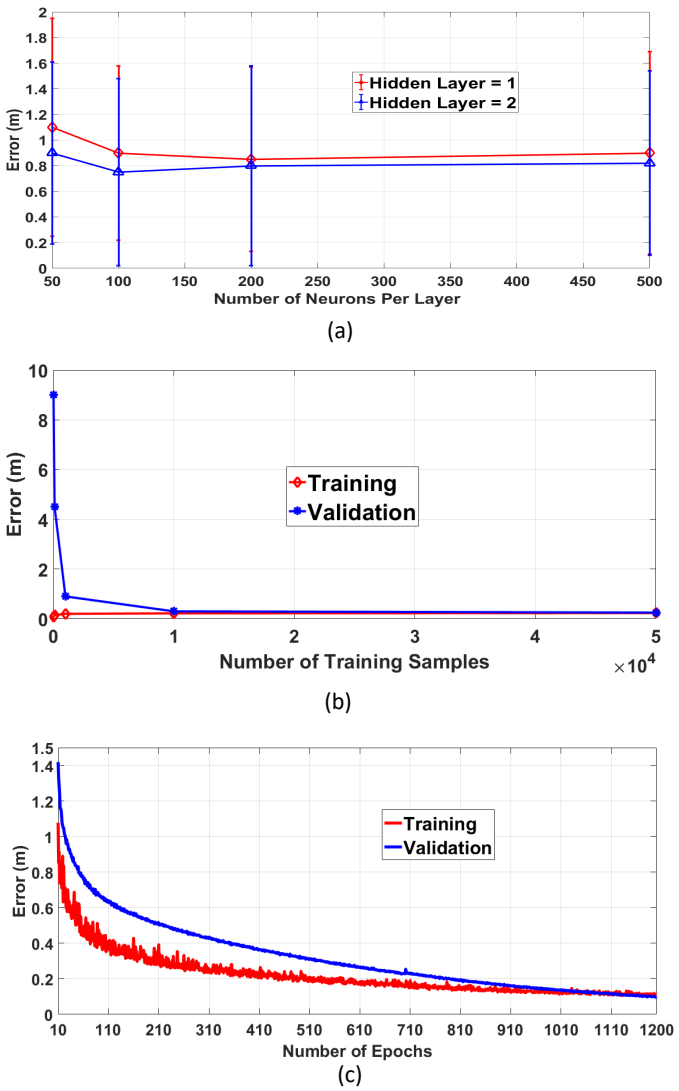


Figure 4.6: (a) Average localization errors of P-MIMO LSTM with different number of hidden layers and neurons per layer. (b) Learning curve of P-MIMO LSTM with the number of training trajectory samples vs the average localization error. (c) Learning curve of P-MIMO LSTM with the number of running epochs vs the average localization error (the training trajectory samples = 10^4).

Number of Training Trajectories and Training Epochs

The learning curve can determine the minimum required number of training trajectory samples and training epochs for the proposed RNN models. Fig. 4.6(b) shows the relationship between the training and validation testing errors vs. the number of

Table 4.4: Different learning rates and optimization algorithms

Optimization algorithm	**Learning rate**	**Average Error (m)**
Adam	0.01	1.0 ± 0.78
Adam	0.001	0.75 ± 0.64
Adam	0.0001	0.80 ± 0.62
SGD	0.01	1.52 ± 1.32
SGD	0.001	1.92 ± 1.52
SGD	0.0001	1.85 ± 1.25
RMSProp	0.01	1.05 ± 0.95
RMSProp	0.001	0.88 ± 0.72
RMSProp	0.0001	0.85 ± 0.68

Table 4.5: Different dropout rates

Dropout rate	**Average Error (m)**
0.1	0.72 ± 0.68
0.2	0.75 ± 0.64
0.3	0.87 ± 0.74
0.4	0.92 ± 0.73

training trajectory samples of the proposed P-MIMO LSTM model. According to Fig. 4.6(b), overfitting will be mitigated if the number of training samples increases to 10^4. Therefore, 10^4 is the minimum number for the random training trajectories to feed to P-MIMO LSTM in the training phase. Fig. 4.6(c) illustrates the relationship between the training and cross-validation errors vs. the number of running epochs of P-MIMO LSTM when the training trajectory samples is 10^4. When the cross-validation error is at a minimum on the graph, the minimum number of epochs is approximately 1,000.

Learning Rate, Optimization Algorithm and Dropout Rate

Table 4.4 shows the localization errors when the learning rates and optimization algorithms are varied. The other parameter settings still follow Table 4.2. Clearly, optimizer ADAM with the learning rate 0.001 provides the best results with the average error being 0.75±0.64 m. Table 4.5 demonstrates the accuracy of different dropout rates. With the dropout rate equals or less than 0.2, the performance of

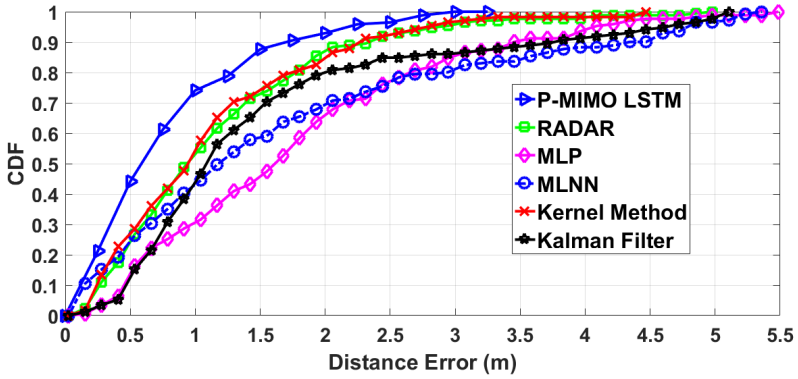


Figure 4.7: The CDF of the localization error of P-MIMO LSTM and the other methods in literature.

P-MIMO LSTM is mostly unchanged, i.e., 0.72±0.68 m at a dropout rate of 0.1 and 0.75±0.64 m at a dropout rate of 0.2. After the dropout rate increase above 0.2, the accuracy is deteriorated and reaches the bottom of 0.92±0.73 m at a dropout rate of 0.4.

4.5.4 RNNs Comparison

Fig. 4.4(d) compares the performance between vanilla RNN [66], LSTM [10], GRU [11], BiRNN, BiLSTM [12] and BiGRU [13]. All of the settings follow Table 4.2. Although the gap between these systems are close, LSTM still consistently has the best performance with the average error at 0.75±0.64 m compared to 1.05±0.77 m of RNN, 0.80±0.70 m of GRU, 0.95±0.77 m of BiRNN, 0.89±0.75 m of BiLSTM and 0.93±0.81 m of BiGRU, respectively. While 80% of the errors of RNN and BiLSTM are all within 1.5 m, the ones of LSTM and GRU are within 1.2 m. Some explanations are as follows.

Regarding vanilla RNN, there is a disadvantage of vanishing gradient at large T [71]. Therefore, RNN has difficulty to learn from the long-term dependency, i.e, $T = 10$ in our case. Unlike the traditional RNN, both LSTM and GRU are able

to decide whether to keep the existing memory from the past by their gates, i.e, forget gate in LSTM and reset gate in GRU. Intuitively, their performances in our experiment are both better than vanilla RNN because if LSTM and GRU detect an important feature from an input sequence at early stage, they easily carry this information over a long distance and capture potential long-distance dependencies. Furthermore, GRU is a simpler version of LSTM, which means that some of the feature of LSTM are reduced in GRU, e.g., the exposure of the memory content control [71]. Therefore, LSTM has a slightly better performance than GRU. For the bidirectional models including BiRNN, BiLSTM, BiGRU, they use both historical and future information, i.e., the initial location and the last location to predict the current location. In this chapter, we assume only the very first location of each trajectory is known. As the ground truth of the last location is not incorporated, error is introduced into the bidirectional models. Therefore, the bidirectional models are not favorable in our work.

4.5.5 Literature Comparison

Fig. 4.7 compares the proposed P-MIMO LSTM with the feed-forward neural network MLP [62], multi-layer neural network (MLNN) [64] and the other conventional methods including KNN-RADAR [21], probabilistic Kernel method [46] and Kalman filter [24]. In our experiments, we simply redo the MLP and MLNN methods in [62,64]. The MLP model has three layers with only one 500-neurons hidden layer. By comparison, MLNN has five layers, i.e., 1 input layer, 3 hidden layers with 200, 200 and 100 neurons respectively per layer and 1 output layer. The input of those memoryless methods is a single RSSI vector (11 RSSI readings) of a specific location, the output is a single location. In opposite, the input of P-MIMO LSTM is a trajectory with T RSSI vectors ($11\times T$ RSSI readings) from T time steps, the output is a trajectory

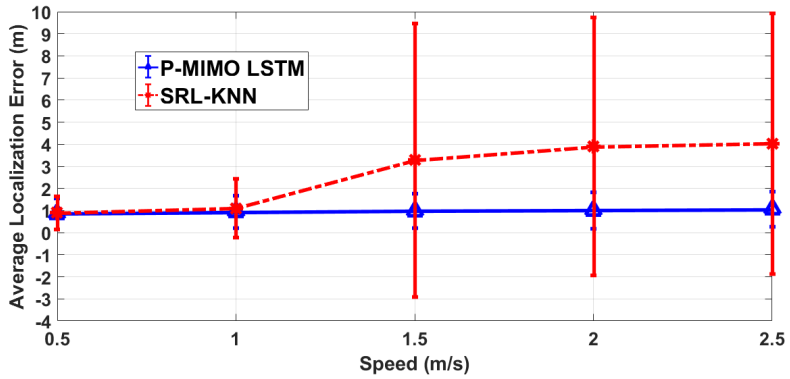


Figure 4.8: Average localization errors with the error bars of P-MIMO LSTM and SRL-KNN in changing speed scenarios

including T output locations. P-MIMO LSTM clearly outperforms MLP with the maximum error of 3.4 m compared with 5.5 m of MLP. 80% of LSTM model errors are within 1.2 m, which is much lower than 2.7 m of MLP. The maximum errors of conventional methods such as RADAR, Kalman filter and Kernel method are more significant 4.8 m, 5.0 m and 4.50 m, respectively. Besides, 80% of the errors of those methods are all within around 2 m, 1.7 times higher than the proposed LSTM model. Table 4.3 lists the average errors between LSTM models and the other mentioned methods. Clearly, the accuracy of P-MIMO LSTM with 0.75 m dominates the other conventional feed-forward neural networks with 1.65 m of MLNN [64] and 1.72 m of MLP [62].

4.5.6 Further Discussion

Three Challenges of WiFi Indoor Localization

As mentioned in Section 4.1, the proposed LSTM models can address three challenges of WiFi indoor localization. Subsection 4.5.3 has demonstrated that LSTM adopts a sequential measurements from several locations in the trajectory and decreases significantly the spatial ambiguity (first challenge). In addition, RSSI instability

Table 4.6: Average localization errors of UJIIndoorLoc database

	P-MIMO LSTM	**RADAR [21]**	**MLNN [64]**	**Kalman Filter [24]**	**MLP [62]**
Building 0 (m)	4.5 ± 2.7	7.9 ± 4.9	7.6 ± 4.2	8.2 ± 5.0	9.2 ± 5.8
Building 1 (m)	4.0 ± 3.8	8.2 ± 4.9	7.5 ± 3.3	8.4 ± 3.9	7.4 ± 4.4
All buildings (m)	4.2 ± 3.2	8.1 ± 4.9	7.5 ± 3.8	8.2 ± 4.7	8.2 ± 5.2

(second challenge) and RSSI short collecting time per location (third challenge) create the diverse values of RSSI readings in one location, which can lead to more locations having similar fingerprint distances. The proposed LSTM models can effectively remove the number of false locations which are far from the previous points based on the series of the previous measurements and predictions. Therefore, the adverse effects of the second and third challenge can be mitigated.

Training Time Requirement

Compared with other conventional methods, e.g., KNN-RADAR [21], probabilistic Kernel method [46] and Kalman filter [24], the proposed RNN method outperforms. However, those conventional methods do not require the training phase, which compares the current RSSI measurement with the ones in the database to get users locations directly. In contrast, the proposed RNN methods require to train the neural network model beforehand. In the training phase, RNN models will adjust the internal weights and use those weights to instantly predict the users location in the testing phase. The learning curve can determine the minimum required training time for the proposed RNN models. From Subsection 4.5.3, the optimal training trajectory samples is 10^4 and the optimal number of epochs is approximately 1,000 for 1 fold test. In the experiment, our training server configuration is AMD FX(tm)-8120 Eight-Core CPU and Nvidia GTX 1050 GPU. For 1 epoch, the running time is approximately 4 s. Therefore, the minimum optimal time for 1 fold test of the proposed RNN network to get the good accuracy is $4 \times 1000 = 4000$ s ($\cong$ 1 hour and 6 minutes).

In practical scenario, the more APs are used, the more RSSI features can be

extracted, leading to the better performance. However, increasing the number of APs also creates more computational cost and extends the training time. [73] suggests to use LSTM with projection layer (LSTMP) to reduce the computational cost by adding a linear projection layer after the LSTM layer. Furthermore, LSTMP was reported helpful to error rate reduction because parameter reduction helps the LSTM generalization. In the future work, when the number of APs are increased to achieve better accuracy, the idea of LSTMP can be applied to reduce the training time of our RNN models.

Changing Speed Scenario

Some conventional short-memory methods such as Kalman filter [24], SRL-KNN [8] have the constraints about the speed of the users. If the users change their speed rapidly, the localization accuracy of those methods will be affected severely. In contrast, the proposed LSTM network is trained with random trajectories as described in Subsection 4.3.2 with no strict constraints about the speed of the users, so it can handle the changing speed scenarios. Fig. 4.8 illustrates the average errors of the proposed P-MIMO LSTM and SRL-KNN using RSSI mean database with parameter $\sigma = 2$ m [8]. The number of testing points are 344 locations following the backward and forward trajectory like Fig. 2.1. The maximum speed is the instant speed of the robot between 2 random consecutive testing locations in a sampling time interval Δt. The number of locations having the maximum speed are 50% of the total testing points (172 locations). The rest of the locations have a random speed smaller than the maximum speed. When the maximum speed increases from 0.5 m/s to 2.5 m/s, the average errors of LSTM model stays stable around 0.85$\pm$0.75 m. On the other hand, SRL-KNN starts from the comparable result as P-MIMO LSTM with 0.90 m in the case of 0.5 m/s. After the maximum speed increases to above 1.5 m/s, the

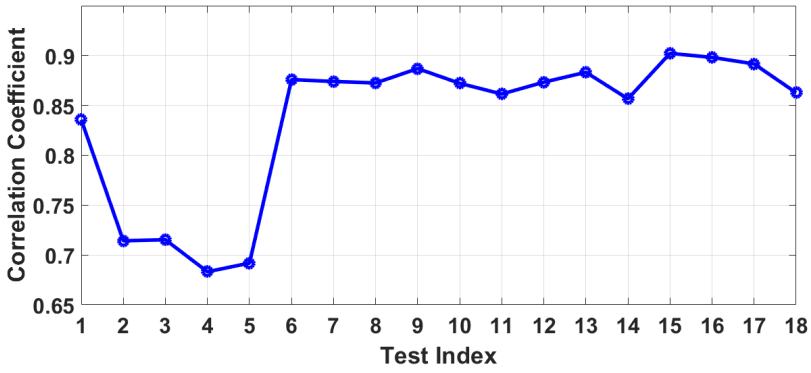


Figure 4.9: Average correlation coefficient between different time trajectory tests and the database.

accumulated errors appear and the accuracy of SRL-KNN is significantly degraded to be above 3 m with the large variation being above 5 m.

Tests At Different Time Slots

The proposed RNN methods first learn the RSSI range characteristics of the environment offline in a prior training phase before using the learned characteristics during the testing phase. Therefore, the initial data in the training phase might not have the same distribution with the data in the testing phase. In our experiment, we address that problem with the support of our autonomous robot as shown in Fig. 4.1(a). The robot is programmed to repeatedly run around the experimental area and collect the new data to update the database in different hours and days. The proposed RNN networks are trained with a wide variety of data reflecting the different characteristics of the environment at different time periods. Fig. 4.9 illustrates the Pearson correlation coefficients between different time trajectory tests and the appropriate neighbour locations in our database. We repeatedly collect the testing trajectory with 175 locations as shown in Fig. 2.1(a) at 18 random hours. The correlation coefficients range from 0.7 to 0.9, which proves that we can always find similarly distributed data in the database corresponding to each single test. Furthermore, according to the survey

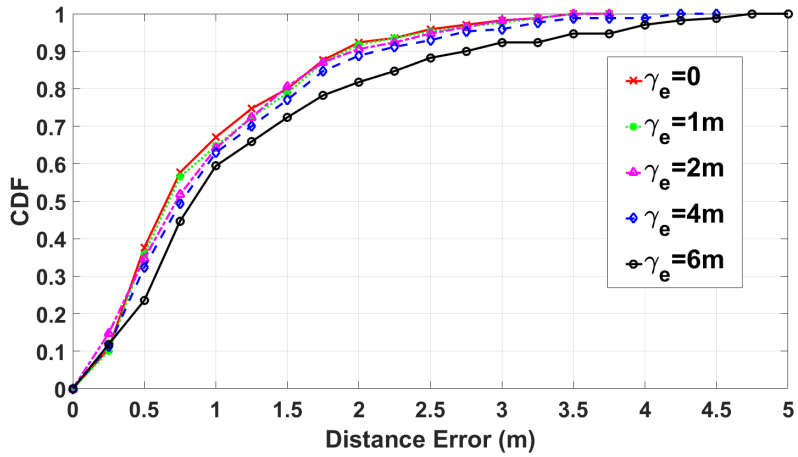


Figure 4.10: CDF of P-MIMO LSTM localization errors in different historical data error scenarios.

in [47], the percentage of stationary time when a user is not moving can exceed 80% for most mobile users. Some of the locations falling in the stationary period can serve as the anchor points to re-calibrate the network in the testing phase [74]. On the other hand, if other sensors such as camera are available, the additional data can help to increase the estimation accuracy of locations in the trajectory [74]. A detailed study of these issues are out of the scope of this chapter but will be addressed in our future work.

Stability and Robustness

Since P-MIMO LSTM leverages the information of a user's previous positions to estimate the current location, the stability of P-MIMO LSTM depends on the accuracy of historical data from the previous steps. In order to investigate the propagation error due to the imperfect prior location estimation, Fig. 4.10 illustrates the localization errors of P-MIMO LSTM with both the ideal and erroneous history data. Starting with the perfect historical coordinate $\boldsymbol{h}(x, y)$ for every location in the testing trajectory, an amount of Gaussian error is added to $\boldsymbol{h}$. The erroneous prior location $\boldsymbol{h'}(x', y')$

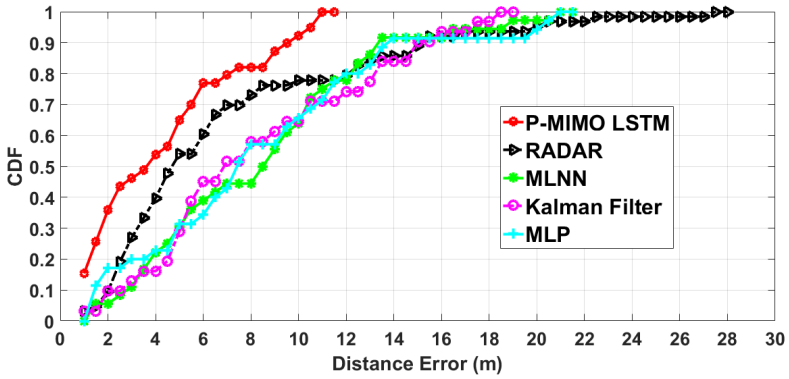


Figure 4.11: Localization error CDF of UJIndoorLoc database for all buildings

is obtained as: $x' = x + x_e$, $y' = y + y_e$, where x_e and y_e are random variables that follow Gaussian distribution

$$x_e \sim \mathcal{N}(0, \sigma_{x_e}^2)\,;\; y_e \sim \mathcal{N}(0, \sigma_{y_e}^2)\,;\; \gamma_e = \sqrt{\sigma_{x_e}^2 + \sigma_{y_e}^2}$$

Fig. 4.10 shows that if the standard deviation error γ_e of the historical data is within 2 m, the localization accuracy is mostly similar to the ideal case, with a maximum error of 3.5 m and 80% of the error is 1.5 m. When γ_e increases to 4 m and 6 m, the accuracy becomes slightly worse with the maximum errors being around 5 m and 80% errors being around 1.80 m and 2.3 m, respectively. As shown in Table 4.3, the average errors of P-MIMO LSTM is within 1 m, i.e., 0.75 ± 0.64 m, which indicates that our proposed P-MIMO LSTM is robust to the localization error of the previous positions.

4.5.7 Other Database Comparison

The consistent effectiveness of the proposed LSTM system is proved by the published dataset, UJIIndoorLoc [48]. The reported average localization error in [48] is 7.9 m. The database from 2 random phone users (Phone Id: 13, 14) in 2 different buildings

(Building ID: 0 and 1) are used to implement P-MIMO LSTM. Note that the grid size of UJIIndoorLoc is different from the collected database and so is the average localization error. However, the relative accuracy comparison between the proposed LSTM and conventional KNN, e.g., RADAR [21] and Kalman filter [24] or feed-forward neural network, e.g, MLP [62] and MLNN [64] can verify the effectiveness of our algorithm. Table 4.6 shows the average errors in meter of P-MIMO LSTM, RADAR, Kalman filter, MLP and MLNN for each separate building and for all 2 buildings in general. For all 2 buildings, the average error of P-MIMO LSTM is 4.2±3.2 m, significantly lower than the result of RADAR 8.1±4.9 m, MLNN 7.5±3.8 m, Kalman filter 8.2±4.7 m and MLP 8.2±5.2 m. Furthermore, Fig. 4.11 compares the CDF of localization errors between those methods. In total, a 11.5 m maximum localization error is recorded for P-MIMO LSTM, 22 m for MLNN and the largest maximum localization error of 28 m for RADAR. Besides, 80% of the error is below 7 m in the case of P-MIMO LSTM, which is much lower than 12 m in the case of MLP, MLNN and RADAR.

4.6 Conclusions

In conclusion, we have proposed recurrent neural networks for fingerprint indoor localization using WiFi. Our RNN solution takes into account the relation between a series of the RSSI measurements and determines the user's moving path as one problem. Experimental results have consistently demonstrated that our LSTM structure achieves an average localization error of 0.75 m with 80% of the errors under 1 m, which outperforms feed-forward neural network, conventional methods such as KNN, Kalman filter and probabilistic methods. Furthermore, main challenges of those conventional methods including the spatial ambiguity, RSSI instability and the RSSI

short collecting time have been effectively mitigated. Besides, the analysis of vanilla RNN, LSTM, GRU and BiLSTM with important parameters have been discussed in details, i.e, loss function, memory length, input and output features, etc.

Chapter 5

A CNN-LSTM Quantifier for Single Access Point CSI Indoor Localization

5.1 Introduction

The previous chapters show that many WiFi fingerprinting localization systems rely on RSSI, which can be obtained from most WiFi receivers such as mobile phones, tablets, laptops, etc., [17]. However, in order to locate the accurate positions of users using RSSI, a large number of access points (APs) are required. For example, 6 APs were used in SRL-KNN [8] and 15 APs in DANN [19]. Because of the rapid changes in indoor environments, such as shop renovations or mall upgrades, a large number of APs for indoor localization purpose are not always feasible [75]. Furthermore, in practical scenario, many small areas have only a single available AP such as in a small store, local pharmacy or classroom, which is impossible to use only RSSI for accurate localization. In order to address that problem, CSI is adopted in this chapter. In

contrast to having only one RSSI per packet, multiple CSI values co-responding to different subcarriers can be obtained at one packet. The rich information in CSI enables localization with a single AP. This chapter explores the algorithm to locate a mobile device using a single WiFi AP in an indoor environment.

CSIs are complex numbers. Therefore several features such as amplitude, phase and time of arrival (ToA) can be utilized. Using CSI amplitude, FILA [58] estimates the signal strength distribution for each AP at each location based on the total power of all CSI subcarriers. After the database is constructed, a probabilistic method with Bayes' rule will estimate the user's location in the testing phase. The experiment shows that FILA achieves a 40% improvement in accuracy compared with RSS based Horus [23]. Because the summation of the subcarriers' power in FILA [58] has the drawback of losing the diversity information, DeepFi [61] proposed a novel deep learning system directly utilizing the CSI amplitudes. In the training phase, the deep learning approach exploits the CSI amplitudes from multiple subcarriers to train the network weights and uses them as fingerprints. In the testing phase, a probabilistic method based on the radial basis function (RBF) is used to obtain the user's location. With only one AP being utilized, DeepFi has the best accuracy of 0.94 ± 0.56 m which outperforms several existing RSS and CSI based schemes. Different from DeepFi, ConFi [5] organizes the CSI amplitudes into a time-frequency matrix that resembles CSI images. These images are the fingerprint features for each location. ConFi models localization as a classification problem and addresses it with a five-layer CNN that consists of three convolutional layers and two fully connected (FC) layers. In [14], the enhanced models of FILA and ConFi with an additional semi-sequential step are proposed to boost up their performance by around 25%.

Beside the amplitude, CSI phase is also widely utilized for localization. For example, PhaseFi [76] first extracts the raw phase value from the complex CSI and

removes the offset through linear transformation to get the calibrated phase. A deep network with three hidden layers is adopted to train the calibrated phase data, and probabilistic Bayes is used for location estimation. Instead of using the coarse phase value directly as in PhaseFi, CIFI [77] extracts the phase of CSI to estimate the phase differences between two adjacent antennae or the angle of arrival (AoA). AoAs are constructed in a form of image to feed to deep convolutional neural networks (DCNN) for indoor localization. The best accuracy of these methods is around 1.0±0.4 m [76]. In BiLoc [7], Wang *et al.* combine both average CSI amplitudes over pairs of antennae and estimated AoAs to form a bi-modal data. A complicated network that incorporates deep autoencoder, restricted Boltzmann machine along with radial basis function is utilized to estimate the user's position. The method is implemented in a total of 25 testing points in two different environments including a computer lab and a corridor, with a best accuracy of 1.5±0.8 m.

Based on CSI, some research works can exploit ToA to locate the user's position by trilateration method [78, 79]. In [78], a set of algorithms named Chronos is proposed to estimate the ToA relying on the phase values after inverse fast Fourier transform (IFFT) of the original CSI. A hopping method between multiple frequency bands is utilized to increase the accuracy of estimated ToA. Based on the experimental results, Chronos achieves 0.47 ns median time-of-flight error, corresponding to a physical localization accuracy of 14.1 cm. Ref. [79] estimates ToA with the super-resolution algorithm [80] by transforming the CSI to time domain pseudospectrum, and TOA being obtained by detecting the first peak of the pseudospectrum in the delay axis. The minimum resolution [79] is 10 ns, corresponding to a physical distance of 3 m.

Up to date, only a few specific wireless network interface cards (NIC) are able to provide CSI readings. These include Intel WiFi Link 5300 MIMO NIC, and devices built on Atheros AR9390 or AR9580 chipset [7]. Recently, Schulz [81] have developed

Table 5.1: Comparisons of indoor localization experiments using CSI

Method	Feature	Access point (AP)	Reference point (RP)	Testing Point	Testing Selection	Accuracy
FILA [58]	Amplitude	1-3	28	-	Fixed	0.4 m to 1 m
DeepFi [61]	Amplitude	1	50	30	Fixed	0.94 ± 0.56 m
ConFi [5]	Amplitude	1	64	10	Fixed	1.3 ± 0.9 m
PhaseFi [76]	Phase	1	38	12	Fixed	1.0 ± 0.4 m
CIFI [77]	Phase	1	15	15	Fixed	1.7 ± 1.2 m
BiLoc [7]	Amplitude & Phase	1	25	25	Fixed	1.5 ± 0.8 m
Proposed CNN-LSTM	Amplitude	1	**1,185**	**195**	**Random**	See Table. 5.5

Nexmon, a firmware patching framework for Nexus 5 smartphone. Nexmon can extract the CSI data from the PHY and send them to the user's interface through UDP frames.

Despite of extensive investigation, some of the following issues still exist in all of the above methods.

1 Previous CSI experiments were limited by the insufficient number of reference points (RPs) and testing points per unit testing area as the measurements on the RPs are conducted manually. Table 5.1 compares the experimental areas and results between different algorithms. The largest number of RPs is 64 in ConFi [5], and that of testing points is 30 in DeepFi [61]. As the localization accuracy is determined by the density of the RPs in the target area, the performances of the reported algorithms are limited. Further, due to the limited number of measurements conducted manually, the previous works select fixed testing points or even on the same spots of the RPs and treat the localization as a classification model, which further departs the model generalization from practical cases although its accuracy appears to be high compared to practical scenarios. A detailed comparison with closer-to-real dataset will be presented in Table 5.5.

2 The influence of CSI fluctuations are essential to localization accuracy but rarely investigated in past research. With the presence of a human, CSI readings from a subchannel can decrease (or increase) significantly [82]. As a result, CSI

fingerprints in the database may not match the instant CSI readings in the testing phase.

3 The information of the previous time steps in user's trajectory has not been exploited to enhance localization. Since the moving speed of the user in an indoor environment is bounded, the historical data from previous steps can provide useful information to predict the current user's location.

To address the first issue, we use an autonomous robot to conduct extensive measurements in different time slots. Since the automated procedure enables us to acquire a lager number of data, there is no need for us to limit the testing point position to be the same as the reference point or at fixed poistions that are a priori known. In our experiments, the testing points are selected anywhere in the target area and estimated through a quantification model. This approximates well to the random walk of a user in a real scenario compared to the previous research. With the sufficient amount of data, not only the changes of CSI through time can be fully analyzed, but also the more comprehensive solutions can be constructed to handle the fluctuation of the CSI. Regarding the last 2 issues, this chapter combines convolutional neural network (CNN) and long-short term memory network (LSTM). CNN extracts the spatial features from several CSI readings, and then LSTM will exploit them in sequential time steps to determine the user's location. These CSI images can be either amplitude, phase or ToA. However, CSI phase is prone to noise and random fading [61]. Therefore, complicated preprocessing is needed before using it as a feature [5]. Besides, it is difficult to get accurate ToA which is only meaningful when the line of sight (LOS) exists. In order to avoid heavy preprocessing, CSI amplitude is a more suitable fingerprint for our proposed method. The proposed algorithm will be tested with the collected CSI from 2 independent devices, i.e, Intel WiFi 5300 NIC card installed in a laptop and Nexus 5 smartphone.

5.2 Proposed Method

5.2.1 Localization Scene Overview

The fingerprinting localization system is generally divided into two phases: a training phase and a testing phase. In the training phase, CSI at each predefined reference point (RP) are collected and stored to a database as fingerprints. Only 1 single AP is used along with M collected RPs in the area of interest. Each RP i at its physical location $\boldsymbol{l}_i(x_i, y_i)$ has a corresponding CSI from multiple antennae and multiple subcarriers. A single CSI is a complex number with amplitude and phase. However, the phase of CSI is noisy due to fading effects and frequency offset [5,61]. Therefore, only CSI amplitude is utilized in this chapter. In each measurement at RP i, we group H number of CSI amplitude measurements with W subcarriers to construct an $H \times W$ matrix

$$\tilde{A}(\boldsymbol{l}_i) = \begin{bmatrix} A_i^{11} & A_i^{12} & \dots & A_i^{1W} \\ A_i^{21} & A_i^{22} & \dots & A_i^{2W} \\ \vdots & \vdots & \ddots & \vdots \\ A_i^{H1} & A_i^{H2} & \dots & A_i^{HW} \end{bmatrix}. \tag{5.1}$$

Here, A_i^{hw} is the CSI amplitude value of the subcarrier w in the measurement h at RP i. Matrix $\tilde{A}(\boldsymbol{l}_i)$ has the form of an $H \times W$ image, which contains the WiFi channel information of location i. Therefore, in the rest of the chapter, $\tilde{A}(\boldsymbol{l}_i)$ is called a CSI image of location i. During the training phase, a large number of CSI images (W_1) are collected at a single RP while in the testing phase, only a small number of CSI images (W_2), are available as the user is expected to frequently moving in practical scenario. In the testing phase, each unknown location of the user, denoted as a testing point, is determined by the localization algorithm.

5.2.2 CNN-LSTM Overview

CNN is a deep learning architecture. It has been successfully used in computer vision and activity recognition [83]. By using convolutional kernels, CNN has been proved to be an effective technique in image classification [84]. There are 3 main types of layers in a typical CNN, including convolutional, max pooling, and fully connected layers. The convolutional layer can obtain useful spatial features from an image, while the max pooling layer shrinks the size of the image to reduce the computational complexity. Finally, the fully connected layer can convert the image to a one-dimension vector to train the data.

RNN is explained in Chapter 4 with the implementation of various types of RNN, including vanilla RNN, LSTM and all their related bi-directional models for indoor localization. Based on the extensive experiments using WiFi RSSI, we concluded in Chater 4 that LSTM provides the best accuracy among all. Therefore, in this chapter, we use LSTM as our sequential model. In the following section we will describe details about our proposed WiFi CSI indoor localization system using the combination of CNN-LSTM model.

5.2.3 Proposed Localization System

The architecture of the proposed CNN-LSTM system is presented in Fig. 5.1(a). The detailed CNN-LSTM model is shown in Fig. 5.1(b). The CSI data for both training and testing will be collected with the support of an autonomous robot. Details of the process will be described as follows.

CSI Filter And Normalization

By using the modified firmware released for Intel WiFi 5300 NIC card [60] and for Nexus 5 smartphone [81], the mobile devices can get one CSI reading per beacon frame

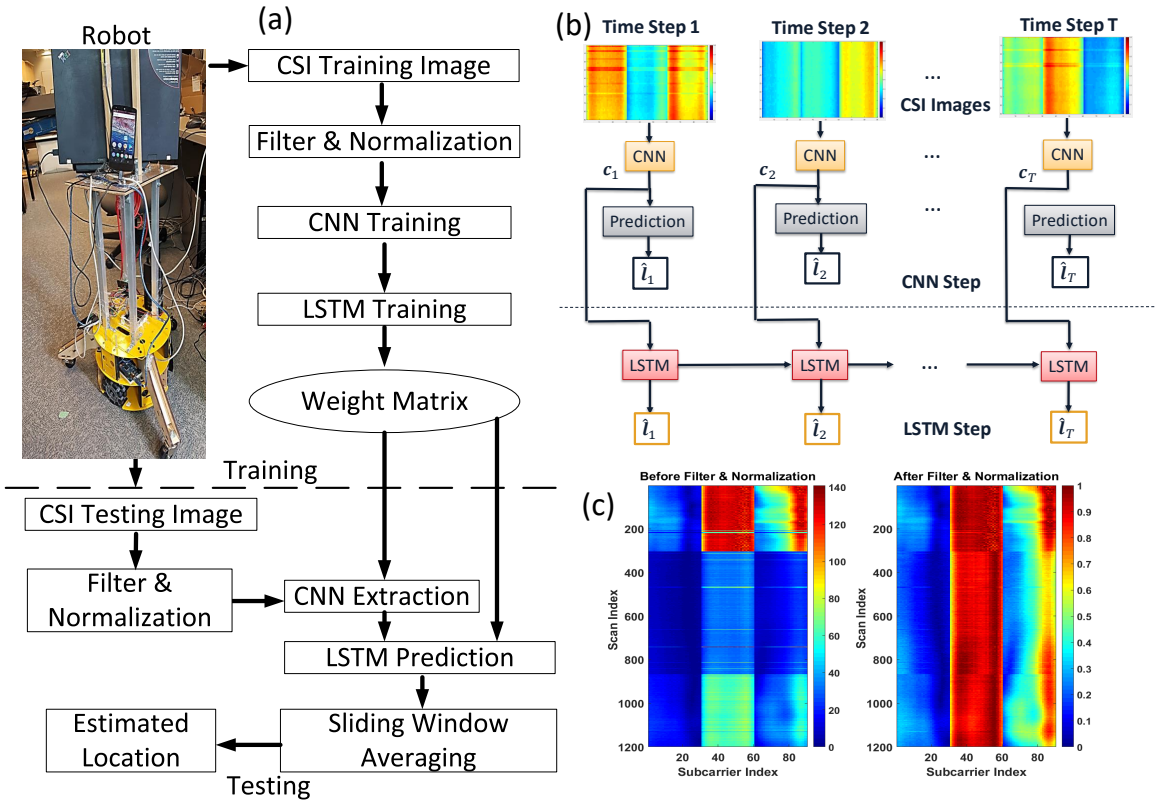


Figure 5.1: (a) Localization process of the proposed CNN-LSTM system. (b) Proposed CNN-LSTM model. (c) CSI images before and after applying filter and normalization.

or data frame. Although this CSI provides rich information from multiple subcarriers, it is sensitive to the change of environments such as human blocking and movements, interference from other equipment and devices, receiver antenna orientation, etc., [82]. In order to filter out the outliers caused by the interference and make the measurement more stable, we adopt the median filter [85] for every subcarrier (every column of an CSI image). Then an important normalization process is applied to get cleaner data from the original image. Firstly, the original average amplitude of each image at each location $\boldsymbol{l}_i$ is calculated as

$$A_i = \frac{\sum_{h=1}^{h=H} \sum_{w=1}^{w=W} A_i^{hw}}{H \cdot W}. \tag{5.2}$$

The set of average amplitude of every location in a training database is saved in an array $\{A_1, A_2, \ldots, A_M\}$. The maximum value in the set is denoted as A_{max} at location $\boldsymbol{l}_{max}$. Secondly, every CSI measurement at every location (row of the CSI image) is normalized by the min-max normalization method. In each row h of the image, among W subcarriers, the minimum amplitude is denoted as

$Min(A_i^{hw}) = min([A_i^{h1}, A_i^{h2}, \ldots, A_i^{hW}])$ and maximum amplitude is denoted as

$Max(A_i^{hw}) = max([A_i^{h1}, A_i^{h2}, \ldots, A_i^{hW}])$. The normalized amplitude of a single pixel at row h, column w of the CSI image at location $\boldsymbol{l}_i$ is given by

$$\hat{A}_i^{hw} = \frac{A_i^{hw} - Min(A_i^{hw})}{Max(A_i^{hw}) - Min(A_i^{hw})}. \tag{5.3}$$

The normalized CSI image at a location will become

$$\tilde{\hat{A}}(\boldsymbol{l}_i) = \begin{bmatrix} \hat{A}_i^{11} & \hat{A}_i^{12} & \ldots & \hat{A}_i^{1W} \\ \hat{A}_i^{21} & \hat{A}_i^{22} & \ldots & \hat{A}_i^{2W} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{A}_i^{H1} & \hat{A}_i^{H2} & \ldots & \hat{A}_i^{HW} \end{bmatrix}. \tag{5.4}$$

Finally, in order to preserve the original power of the image, the CSI image at each location $\boldsymbol{l}_i$ is normalized again with the pre-calculated average amplitude set given by

$$\tilde{\hat{A}}(\boldsymbol{l}_i) = \tilde{\hat{A}}(\boldsymbol{l}_i)\frac{A_i}{A_{max}}. \tag{5.5}$$

Fig. 5.1(c) illustrates an example of CSI images collected by Intel WiFi 5300 NIC card before and after applying the filter and normalization process. There are noisy readings (scattered rows), caused by the fluctuation of the CSI amplitude, inside the original CSI image. However, after the median filter and normalization process, the quality of CSI image is improved with more consistent amplitude. The more detailed

performance analysis will be discussed more in Subsection 5.4.

Proposed CNN-LSTM Model

The proposed CNN-LSTM architecture is trained by the CSI images from consecutive locations in a trajectory to exploit the time correlation between them. Each location in a trajectory appears in one different time step. The length of a trajectory, or equivalently, the number of time steps, defines the memory length T as illustrated in Fig. 5.1(b). Because all weights and hidden state values will be saved in every time step in a training trajectory [70], the number of time steps T impacts the performance of LSTM and CNN-LSTM model. Larger T reveals more information from the past but accumulates more errors on prediction [59]. Under the constraints that the distance between consecutive locations is bounded by the maximum distance a user can travel within the sample interval in practical scenarios, the CNN-LSTM training trajectories are randomly generated based on the method proposed in [59].

There are two main layers in our CNN-LSTM model (Fig. 5.1(b)) including CNN and LSTM layers. While the input of CNN layer is the CSI image from locations in a trajectory, the input of LSTM layer is the output of the previous CNN layer. In other words, CNN will extract the CSI information from spatial domain, whereas LSTM will further exploit that information in the temporal domain. With the information from both domains, the ambiguity of CSI data is significantly mitigated. Subsection 5.4 will show experimental results to demonstrate this mitigation.

The objective of CNN-LSTM training is to minimize the loss function $\mathcal{L}(\hat{\boldsymbol{l}}, \tilde{\boldsymbol{l}})$ which penalizes the Euclidean distance between the output $\hat{\boldsymbol{l}}$ and the target $\tilde{\boldsymbol{l}}$. Then the backpropagation algorithm uses the chain rule to calculate the derivative of the loss function $\mathcal{L}$ and adjusts the network weights by gradient descent [70]. In [86], Valente *et al.* show that CNN-LSTM model should be trained separately, instead of

using only one common output and one loss function. Therefore, the training of our proposed network is divided into two phases. In phase 1, the CNN layer is trained as a quantification model with CSI images in the database and the loss function mean square error (MSE) described as

$$\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}}) = ||\boldsymbol{l} - \tilde{\boldsymbol{l}}||_2. \tag{5.6}$$

The output of the final FC layers in CNN ($\boldsymbol{c}_T$) is extracted and fed to the input of LSTM (Fig. 5.1(b)). In phase 2, the trajectory of CNN output spatial features, $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T\}$, from T time steps is used for LSTM training. The loss function of this phase is still MSE but for T time steps following MIMO-LSTM model [59] as

$$\mathcal{L}(\hat{\boldsymbol{l}}, \tilde{\boldsymbol{l}}) = \frac{\sum_{i=1}^{T} ||\hat{\boldsymbol{l}}_i - \tilde{\boldsymbol{l}}_i||_2}{T}. \tag{5.7}$$

5.3 Database And Experiments

All experiments have been carried out again on the third floor of Engineering Office Wing (EOW), University of Victoria, BC, Canada. The dimension of the area is 21 m by 16 m. It has three long corridors as shown in Fig. 5.2(a). The CSI fingerprints for both training and testing are collected using an autonomous driving robot. The 3-wheel robot (Fig. 5.1(a)) has multiple sensors including a wheel odometer, an inertial measurement unit (IMU), a LIDAR, 3 sonar sensors and a color and depth (RGB-D) camera. It can navigate to a target location within an accuracy of 0.21±0.02 m.

During the experiments, the robot carried the mobile devices (a Google Nexus 5 smartphone and a Thinkpad T520 laptop equipped with an Intel 5300 NIC) to collect CSI. There is only 1 AP (TPlink AC1750) in the experimental area, operating in channel 36 with a 5 GHz frequency band. Fig. 5.2(b) illustrates the heat map

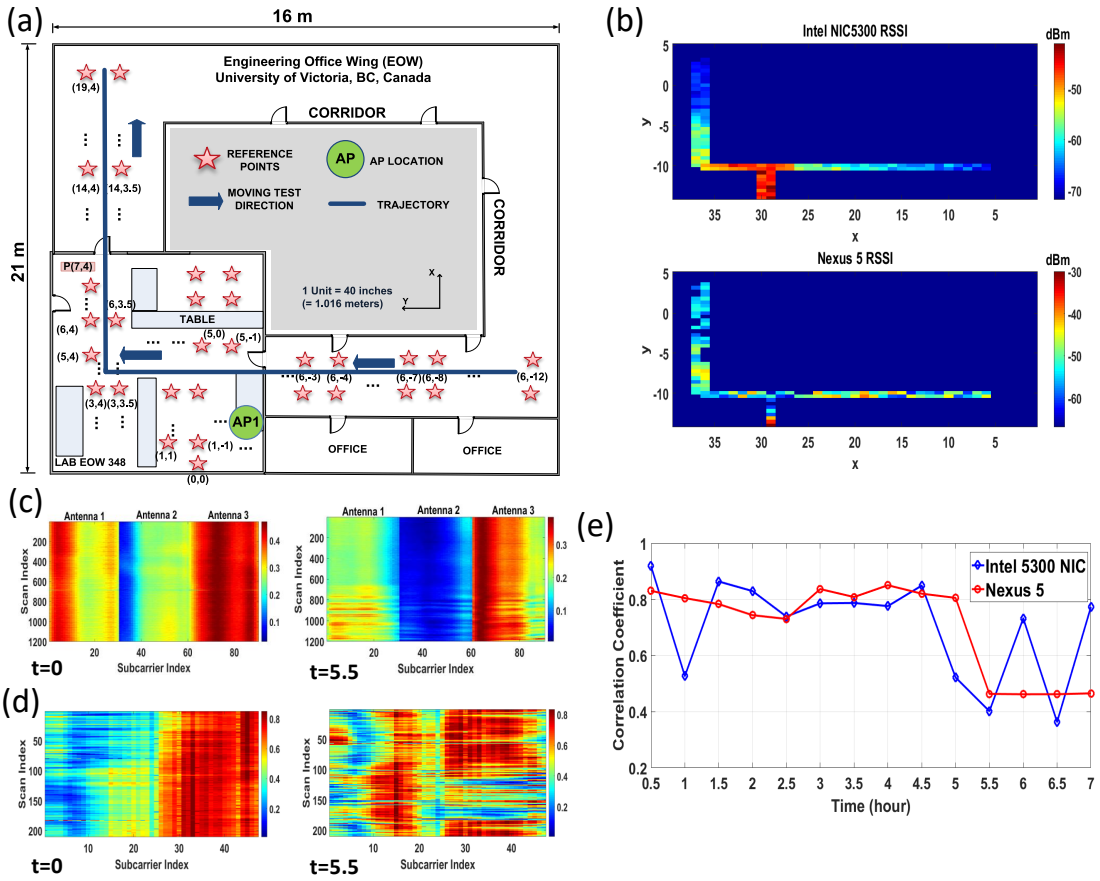


Figure 5.2: (a) Floor map of the CSI test site. The solid blue line is the testing trajectory with blue arrows pointing toward walking direction. (b) Heat map of AP RSSI signal collected from Intel 5300 NIC and Nexus 5 phone. (c) Collected CSI image from Intel 5300 NIC at location (0,0) in 2 different time. (d) Collected CSI image from Nexus 5 phone at location (0,0) in 2 different time. (e) Correlation coefficient of the collected CSI images at location (0,0) along 7 hours.

of the RSSI collected from the Intel 5300 NIC and Nexus 5 smartphone, where the signal strength is represented by color. Clearly, the signals are stronger in the area near the AP. The targeted AP can cover the area including one room (EOW 348) and two corridors with 1185 RPs (pink stars) and 195 testing points (solid blue line). There are 3 receiving antennae on the Intel 5300 NIC, while the smartphone has only one receiving antenna. Therefore, as shown in Fig. 5.2(b), the RSSI signals from the Intel card are stronger and more consistent than the ones from the smartphone.

In the training phase, the robot stays at every RP to collect data for 2 minutes. The CSI data is repeatedly collected in different days and at different time to build the database. In the testing phase, the robot navigates along a pre-defined route (Fig. 5.2(a)) at a speed randomly varying within 0.6-4.0 m/s to simulate the indoor walking pattern of a normal person [40, 41]. There are 195 testing points in each testing trajectory. The test experiment is conducted several rounds per day and repeated in 3 different days. Day 1 test mainly targets the quiet time when the area has fewer people (before 9 am and after 5pm). In contrast, during day 2 and 3 we test the localization accuracy at steady time (working hours) and busy time when more people moving around. Note that the validation data is collected at different time from the training data but at the same RP locations, while testing point locations are randomly selected to be different from all RPs in order to approximate our model closer to practical situations. The user location is updated every Δt =1-2 s.

5.4 Results And Discussions

5.4.1 CSI Sensitivity

The collected CSI data of each antenna of the Intel 5300 NIC include the IEEE 802.11n channel matrices for 30 subcarrier groups, which is about one group for every two subcarriers at 20 MHz bandwidth [60]. Fig. 5.2(c) shows some examples of CSI image in RP location (0,0) from this NIC with 1200 scans and 30 subcarriers for each antenna (total 90 subcarriers). Using Nexmon modified firmware [81], we obtain 64 IEEE 802.11ac-subcarriers at bandwidth 20 MHz with 47 of them providing non-zero CSI data from the only one receiving antenna on the Nexus 5 phone. Fig. 5.2(d) illustrates the CSI images from the phone in the same location (0,0) with 200 scans and 47 non-zero subcarriers.

As mentioned in Subsection 5.2.3, although CSI provides rich information from multiple subcarriers, it is sensitive to the change of environments especially the interference of human blocking and movements. With the presence of a human, CSI measurement from a subchannel can decrease (or increases) widely [82]. Fig. 5.2(e) illustrates the fluctuation of CSI data collected from both Intel 5300 NIC and Nexus 5 phone in RP location (0,0) over 7 hours. The experiment was conducted in working hours when many students (up to 10) used WiFi and moved around the area from time to time. CSI images of both devices were sampled every half an hour and were compared with the original CSI images collected at the start time ($t = 0$ h) using Pearson coefficient [8]. Fig. 5.2(e) shows that CSI data of both devices are stable during the long period of time from $t = 1.5$ h to $t = 4.5$ h, with high correlation coefficients (around 80%). It is the period when students in the testing area mostly sat down and studied. However, during the break time, when people started moving around, CSI data changed significantly with the drop to 40% after 5 hours. Fig. 5.2(c), (d) illustrate the visualization of CSI images at $t = 0$ h and $t = 5.5$ h. We observe significant change in the pattern of the images (the order of strong and weak subcarriers) in both Intel 5300 NIC and Nexus 5. Therefore, CSI fingerprints in the database might not match with the instant CSI readings in the testing phase because they were collected at different time.

According to our knowledge, this important problem has been neglected in most of the WiFi CSI localization research. This chapter is the first one that proposes a feasible solution using an autonomous robot to do the extensive experiments in different time slots. With the sufficient amount of collected training data, the mismatch between database and testing data is significantly reduced.

Table 5.2: CNN Layer Parameters

Category	**Intel NIC**	**Nexus 5**
Training Data	47,400 CSI images	35,000 CSI images
Validation Data	23,700 CSI images	14,000 CSI images
Conv Layer 1	Input($30 \times 30 \times 3$)	Input(10×47)
	5×5 kernels, 10 filters	5×5 kernels, 10 filters
Conv Layer 2	Input($30 \times 30 \times 10$)	Input($10 \times 47 \times 10$)
	5×5 kernels, 10 filters	5×5 kernels, 10 filters
Conv Layer 2	Input($30 \times 30 \times 10$)	Input($10 \times 47 \times 10$)
	5×5 kernels, 10 filters	5×5 kernels, 10 filters
FC Layer 1	9000 neurons	4700 neurons
FC Layer 2	900 neurons	470 neurons
Training Output	Location $\tilde{\boldsymbol{l}}$(x,y)	Location $\tilde{\boldsymbol{l}}$(x,y)
Loss Function	MSE	MSE

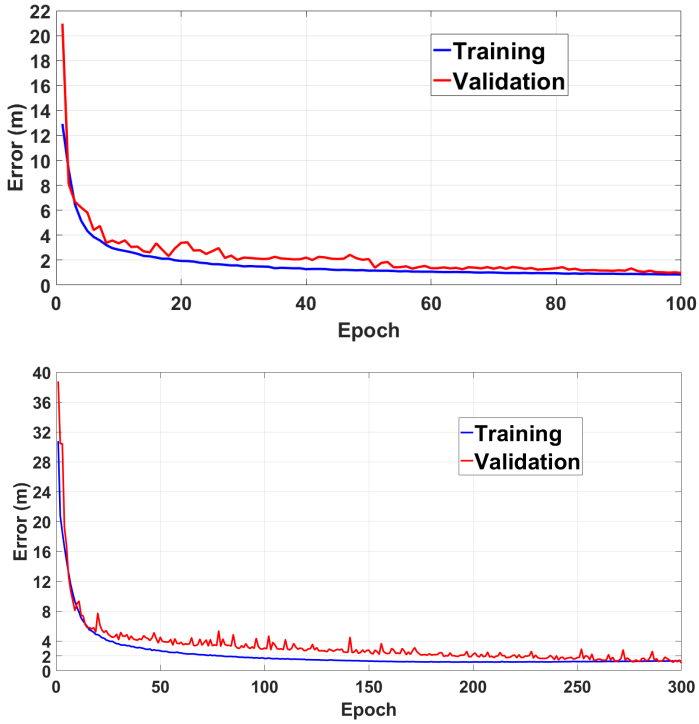


Figure 5.3: CNN Layer Learning Curve (a) Intel 5300 NIC. (b) Nexus 5 Phone.

5.4.2 CNN Layer Analysis

As explained in Section 5.3, the user's location will be updated every Δt =1-2 s. Therefore, the CSI data collected in the window time Δt =1-2 s is utilized for CNN layer training and validation. During Δt, the NIC receives an average of 30 packets,

Table 5.3: Initial setup parameters for RNN system

Category	**Intel NIC**	**Nexus 5**
Memory length (T)	5	5
Model	MIMO-LSTM	MIMO-LSTM
Loss function	MSE	MSE
Hidden layer	900 neurons	470 neurons
Dropout	0.2	0.2
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Training Data	30,000 trajectories	30,000 trajectories
Validation Data	15,000 trajectories	15,000 trajectories
σ	2m	2m
Δ_t	1s	1s

and each CSI packet has total 90 subcarriers from 3 receiving antennae. We consider the data from each individual antenna as a channel of the CSI image, which creates the testing CSI image size for the NIC as $(30 \times 30 \times 3)$. On the other hand, the smartphone only receives 10 scans during Δt with only one single receiving antenna and 47 subcarriers. Therefore, the CNN training CSI image size for Nexus 5 is (10×47).

The CNN model is constructed based on ConFi [5] with 3 convolutional layers (Conv) and 2 FC layers. Table 5.2 shows the detailed parameters of the proposed CNN structure for both NIC and smartphone. The differences between them are the inputs of each layer and the number of neurons for the FC layers. There are 9,000 and 900 neurons respectively in the two FC layers for the Intel NIC, while 4,700 and 470 neurons in the case of using the Nexus 5 smartphone.

The number of training data for CNN layers are 47,400 images for the NIC and 35,000 images for the smartphone. There are 23,700 and 14,000 images respectively for the validation. Fig. 5.3(a) and 5.3(b) illustrate the learning curves of CNN training for the NIC and smartphone data. Both training processes converge at the localization error around 1.7 m. For Intel 5300 NIC data, the proposed CNN model takes 100 epochs, while there are 300 epochs for the smartphone.

After training, CNN outputs spatial features, $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T\}$, of T CSI images

Table 5.4: Average localization errors of CNN-LSTM

	Intel NIC	**Nexus 5**
Day 1 - Quiet Time (m)	2.3 ± 1.7	2.8 ± 2.1
Day 2 - Steady Time (m)	2.7 ± 1.7	3.1 ± 1.7
Day 3 - Busy Time (m)	2.6 ± 1.6	3.1 ± 2.2
Average (m)	2.5 ± 1.6	3.0 ± 2.0

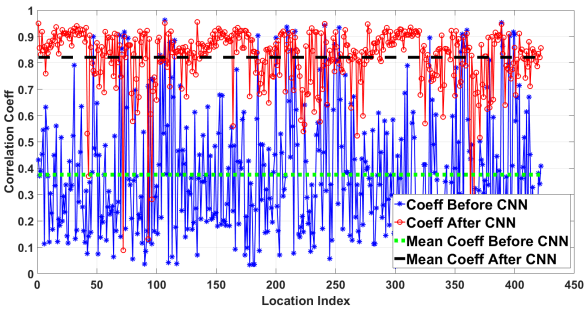


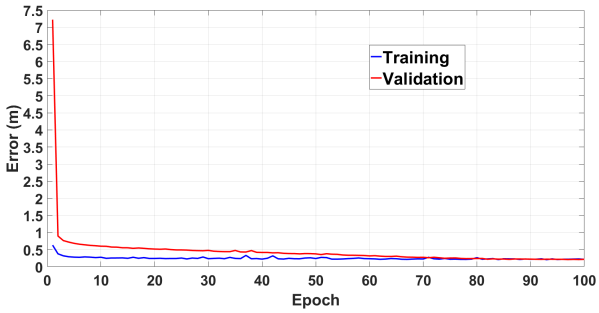
Figure 5.4: Correlation coefficient of original CSI images before CNN and output spatial features after CNN with Intel 5300 NIC dataset.

from T time steps, which are extracted from the output of the last FC layers (FC layer 2). From every CSI image, there are 900 output spatial features for the NIC and 470 features for the smartphone.

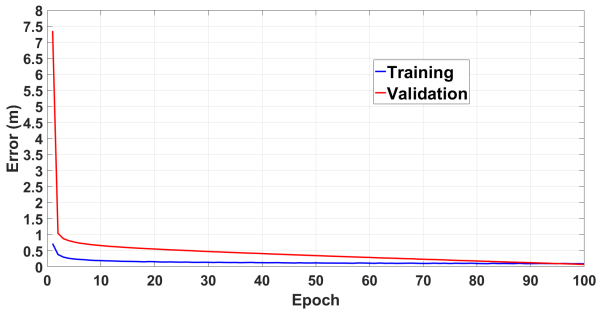
Fig. 5.4 shows the significant increase of the CSI data correlation in 422 random RP locations after CNN layer. Each RP includes 120 CSI images collected at different time. The average correlation coefficient $\rho(\boldsymbol{l}_i)$ at each location $\boldsymbol{l}_i$ is defined as

$$\rho(\boldsymbol{l}_i) = \frac{\sum_{j=1}^{N} \sum_{k=1}^{N} \rho_{jk}(\boldsymbol{l}_i)}{N^2}. \tag{5.8}$$

where $N = 120$ is the number of CSI images at location $\boldsymbol{l}_i$, $\rho_{jk}(\boldsymbol{l}_i)$ is the Pearson coefficient between the j^{th} image and k^{th} image of that location. Before CNN, due to the CSI sensitivity explained in Subsection 5.4.1, the correlation between the original CSI images at different time but the same location is as low as around 40%. However, after CNN, the important spatial features are extracted from the original images with the irrelevant information being removed. Therefore, the correlation between



(a)



(b)

Figure 5.5: LSTM Layer Learning Curve (a) Intel 5300 NIC. (b) Nexus 5 Phone.

output spatial features increases to around 80%. Similar results are obtained from the smartphone database. The CNN output spatial features, $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T\}$, of T will be fed to the next LSTM layer for further processing in time domain.

5.4.3 LSTM Layer Analysis

The detailed parameters for our LSTM network is shown in Table 5.3. The LSTM model follows MIMO-LSTM proposed in [59]. The inputs are multiple generated trajectories containing CNN spatial features from T time steps. Based on the trajectory generation method [59], 30,000 training and 15,000 validation trajectories are generated randomly under the constraints that the distance between consecutive locations is bounded by the maximum distance (σ) a user can travel within the sample interval $\Delta_t = 1$ s. In the conducted experiments in [59], the authors show that the memory length $T = 5$ and the other longer memory cases including $T = 10$ and $T = 40$

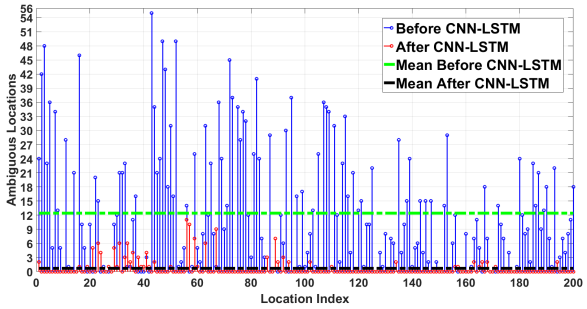


Figure 5.6: Ambiguous locations in the database before and after CNN-LSTM process with Intel 5300 NIC dataset.

provide comparable results. Therefore, the memory length here is chosen as $T = 5$. The outputs include T time steps locations $\{\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_T\}$.

Fig. 5.5(a) and 5.5(b) show the learning curves of LSTM training for Intel 5300 NIC and Nexus 5 smartphone data. Both training processes converge at a localization error around 0.3 m when the number of epochs reaches 100. It is also the localization accuracy of the proposed method in the case when all of the testing points are the same as reference points. However, in the real scenario, due to the random walk behavior of users, that ideal case rarely happens. Therefore, in order to approximate our method closer to practical situations, all of the reported results below are based on the case when testing points are randomly selected whose positions are considered to be a priori unknown.

5.4.4 Performance Analysis

The efficiency of CNN-LSTM layer with both space and time information is evident from the reduction of the ambiguous locations in the database before and after CNN-LSTM training presented in Fig. 5.6. A location $\boldsymbol{l}_j$ is defined as an ambiguous point of $\boldsymbol{l}_i$ if their physical distance is larger than the experimental grid size but their fingerprints have high Pearson correlation coefficient above the correlation threshold.

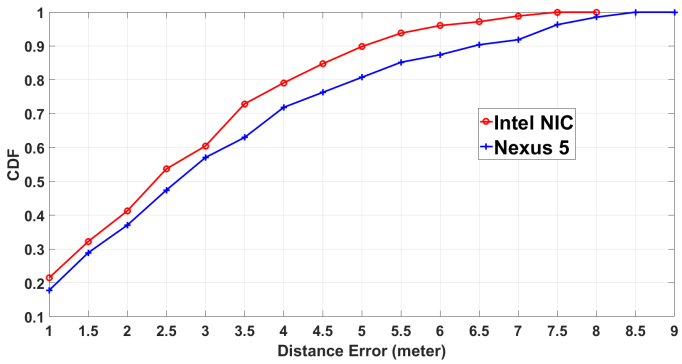


Figure 5.7: The CDF of the localization error of the proposed CNN-LSTM with Intel NIC and Nexus 5 phone

In our experiment, the grid size is 0.5 m. Furthermore, the correlation threshold is chosen based on the average correlation coefficients between $\boldsymbol{l}_i$ and all of its physical nearest neighbours, which is approximately 0.8 in our database. Then all non-nearest-neighbour locations whose correlation coefficient above this threshold are considered as ambiguous points. The fingerprints before CNN-LSTM process is the original CSI image, while the fingerprints after CNN-LSTM is the spatial output feature in $T = 5$ time steps. Pearson correlation coefficient for different fingerprints is explained in details in [59]. Fig. 5.6 shows the number of ambiguous locations among random 200 RPs of Intel 5300 NIC database. Before CNN-LSTM process, there are many locations sharing similar fingerprints (CSI images) with the average of ambiguous locations around 12. The maximum number of ambiguous locations can be up to more than 55. On the other hand, after extracting spatial and temporal information from the original CSI images, the ambiguity reduces significantly. After CNN-LSTM process, more than 80% of the locations do not have any ambiguous point, which proves that their fingerprints become more distinguishable and unique. There are still few locations having ambiguity but the maximum number of ambiguous locations are only 10 points, which is 5 times smaller than the case before the CNN-LSTM process.

Table 5.5: Average Localization Errors - Intel 5300 NIC

	CNN-LSTM	**BiLoc [7]**	**ConFi [5]**	**FILA [58]**
Day 1 - Quiet Time (m)	2.3 ± 1.7	3.7 ± 2.6	6.0 ± 3.8	5.2 ± 2.7
Day 2 - Steady Time (m)	2.7 ± 1.7	4.3 ± 3.0	6.2 ± 3.5	5.9 ± 3.2
Day 3 - Busy Time (m)	2.6 ± 1.6	3.7 ± 2.9	5.7 ± 3.2	5.4 ± 2.8
Average (m)	2.5 ± 1.6	3.9 ± 2.8	5.9 ± 3.5	5.5 ± 2.9

After the training, CNN-LSTM will be utilized for realtime localization tests. Both the laptop with Intel 5300 NIC and Nexus 5 phone are mounted on the robot which moves at the speed 0.6-4.0 m/s following the testing trajectory in Fig. 5.2(a). Every Δt =1-2 s, the server will receive CSI packets sending from both Intel NIC and Nexus phone, to predict the current location. Fig. 5.7 shows the CDF errors of the real testing results. Note again that all of the testing locations are placed randomly and different from all RPs. There are in total 195 collected testing points in each testing trajectory from continuously 3 different days in different time as explained in Section 5.3. The CDF curves in Fig. 5.7 are the combination errors of all those tests. The Intel 5300 NIC has better performance than the Nexus 5 phone with 80% of the localization error below 4 m compared with 5 m of the smartphone. The maximum localization error of the smartphone can be up to 9 m, while the Intel NIC's maximum localization error is only around 8 m. Table 5.4 shows the average localization errors of both devices in 3 days of tests. The accuracy of the Intel NIC consistently outperforms the Nexus 5 phone with the average error being around 2.5±1.6 m and 3.0±2.0 m respectively. The main reason that makes the difference in the performance is the number of antennae. The Intel NIC has 3 antennae, while the Nexus 5 smartphone only has a single antenna. Therefore, the CSI information containing in Intel NIC is more than in Nexus 5 phone, which leads to the more accurate localization results.

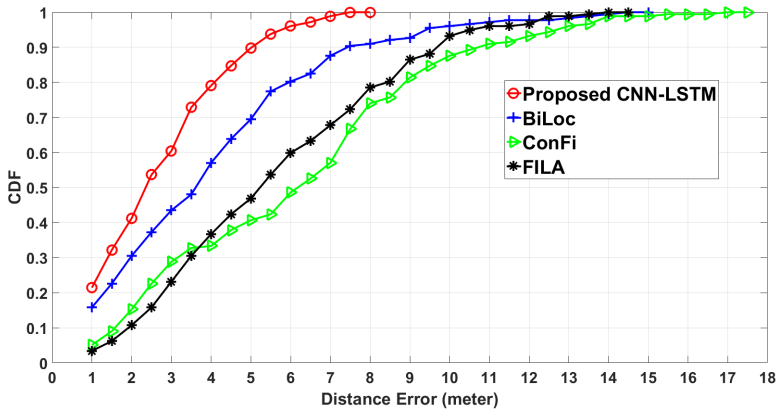


Figure 5.8: The CDF of the localization error of the proposed CNN-LSTM and the other methods in literature.

5.4.5 Results Comparison

In order to confirm the effectiveness of our CNN-LSTM model, the proposed method is compared with 3 other popular methods using CSI fingerprint including FILA [58], ConFi [5] and BiLoc [7]. All of those models are implemented under the same test environment as explained in Section 5.3. Both FILA and ConFi leverage CSI amplitude. FILA uses the probabilistic method with Bayes' rule to estimate the user's location, while ConFi adopts a CNN model with 5 layers to classify the targeted location. In contrast, BiLoc [7] uses both estimated AoAs and average amplitudes as input data. The mixed network, including deep autoencoder, restricted Boltzmann machine along with radial basis function, is used to get the bi-modal fingerprints. Finally, the user's location is determined by the probabilistic model with Bayes' rule. In their research, the CSI data are obtained from Intel 5300 NIC. Therefore, as a fair comparison, all of the latter figures and tables are presented with Intel 5300 NIC database results.

Table 5.5 shows the average error comparison between the proposed method and the three others. All experiments are conducted in 3 days with 195 testing locations, and the proposed model consistently has dominant performance compared with all other 3 approaches. CNN-LSTM has an average error around 2.5±1.6 m, nearly

2.5 times lower than ConFi and FILA whose errors are 5.9±3.5 m and 5.5±2.9 m, respectively. Furthermore, the accuracy of our proposed CNN-LSTM is also 50% better than BiLoc which has an average error of 3.9±2.8 m. Besides, Fig. 5.8 illustrates the CDF error comparison of all four methods. CNN-LSTM with the information of both space and time reduces the maximum error significantly, from more than 17 m of ConFi and 15 m of BiLoc to only 8 m of the proposed approach. Further, 80% of the localization error of CNN-LSTM is below 4 m, which is much lower than that of BiLoc with 6 m and around 8 m of both ConFi and FILA.

5.5 Conclusions

In conclusion, we have proposed a combined CNN-LSTM quantification model for accurate single router WiFi fingerprinting indoor localization. Our CNN-LSTM network extracts both spatial and temporal information of received CSI signals to determine a user's moving path. The extensive on-site experiments, with several mobile devices including mobile phone (Nexus 5) and laptop (Intel 5300 NIC) in hundreds of testing locations, have been presented. The experimental results have consistently demonstrated that our CNN-LSTM structure achieves an average localization error of 2.5 m with 80% of the errors under 4 m, which outperforms the other algorithms by at least 50% under the same test environment.

Chapter 6

Practical Passive Indoor Localization with WiFi Fingerprints

6.1 Introduction

WiFi localization systems so far can be categorized into 2 types, i.e., active and passive classes [16]. All of the previous chapters and the majority of literature work follow the active model which collects WiFi signal from mobile device to infer the user's positions [8, 14]. This active localization class requires a dedicated software installed on mobile devices to perform WiFi scanning and logging data for the localization process [15]. On the other hand, in passive localization, there is no installed software needed in user's devices to obtain fingerprint data. Instead of that, the localization process is fully implemented based on the information collected from WiFi access points (APs) [16, 17].

Because of its practicability, recently, WiFi passive indoor localization has been

further investigated. In some research work, the user's location can be determined without the carry-on devices (device-free). Based on extracted channel state information (CSI) fluctuation from multiple APs induced by the presence of human movement, Shi *et al.* [82] suggests a probabilistic scheme to track user's location. Sharing similar idea but using received signal strength indicator (RSSI) instead of CSI, Liu *et al.* [17] can locate the position by matching the RSSI testing values and the distribution properties stored in the database. Although the above device-free passive localization methods seem to be convenient for users, the experimental area is normally small and their accuracy is not high due to the unpredictable effects of human interference with the original WiFi patterns [87]. In order to have better localization resolution, recent researches tend to focus more on the scenario in which the user carries mobile devices. Duan *et al.* [88] exploits data rate information of all available APs as the utilized fingerprint with the assumption that the data rates at different locations are different. In order to overcome the limitation of the low resolution of WiFi data rate, their subsequent work [15] leverages other fingerprint, namely, packet delivery ratio (PDR). PDR is defined as the proportion of request-to-send (RTS) packets sent by an AP and the number of replied clear-to-send (CTS) frames sent by the mobile devices. Furthermore, a special scheme with lowest-power and highest-modulation is designed for APs transmission to achieve device-invariant. Their method are tested with multiple mobile devices and has the average accuracy 2.5 m.

6.1.1 Device Heterogeneity

Signal strength heterogeneity is the primary problem of indoor localization in general [89]. Because of the difference in power transmission and antenna design, mobile devices can have different signal strength values, even they are in the same location [1].

If the WiFi device used in the training phase is different from the target device in the testing phase, the localization accuracy might be significantly affected [15]. Therefore, there are several methods proposed to address heterogeneous problem. Some of literature work tries to find special fingerprints which do not depend on the device types, i.e., data rate [88], RSS ratio [90], RSSI difference [91]. However, in the real scenario, under the power control mechanism, the number of frames or power sent from a mobile device may vary significantly, which makes all of those fingerprints are less reliable. On the other hand, other works propose the way to calibrate RSSI from 2 different devices. Ref. [89] maps 2 different signal strengths from 2 different devices following linear model. However, in real situation, linear model might not always work due to the non-linear characteristic of WiFi signal in fading environment. In order to improve the calibration performance, Ref. [92] takes into account the RSSI distribution rather than each single RSSI value. Kernel density function is used to reduce the difference between 2 RSSI distributions from 2 different devices. The test is conducted with 3 laptops and 2 phones in 18 RPs with the average accuracy being around 2 m.

6.1.2 Indoor Localization Experiments

Table 6.1 summarizes the indoor localization experimental setup and results. The number of APs, RPs and testing points vary among different experiments, the grid size is defined as the average distance between two adjacent RPs. Among all the indoor localization experiments, the largest number of APs is 15, which is presented in DANN [19]. The largest number of RPs and testing points are 207 and 50 respectively, which is presented in [62], whereas most experiments incorporates less than 100 RPs with maximum 6 APs.

Table 6.1: Indoor localization experimental schemes

Method	Fingerprint	Class	Access point (AP)	Reference point (RP)	Testing Point	Grid Size	Accuracy
FILA [58]	CSI	Active	1-3	28	-	-	0.4 m to 1 m
DeepFi [61]	CSI	Active	1	50	30	-	0.94 ± 0.56 m
ConFi [5]	CSI	Active	1	64	10	-	1.3 ± 0.9 m
PhaseFi [76]	CSI	Active	1	38	12	-	1.0 ± 0.4 m
CiFi [77]	CSI	Active	1	15	15	-	1.7 ± 1.2 m
BiLoc [7]	CSI	Active	1	25	25	1.8 m	1.5 ± 0.8 m
MLP [62]	RSSI	Active	6	207	50	1.7 m	2.8 ± 0.1 m
DANN [19]	RSSI	Active	15	45	46	2 m	2.2 ± 2.0 m
RELM [63]	RSSI	Active	8	30	10	3.5 m	3.7 ± 3.4 m
MLNN [64]	RSSI	Active	9	20	20	1.5 m	1.1 ± 1.2 m
DR Fingerprinting [88]	DR	Passive	4	18	47	5.9 m	3.4 m
PDR Fingerprinting [15]	PDR	Passive	8	60	-	3.3 m	2.6 m

6.1.3 Limitations of existing work

To summarize, although having been extensively investigated in the literature, all of the above approaches have some limitations as follows.

1 All of the active localization approaches require a dedicated software installed on mobile devices to perform WiFi scanning and logging data for the localization process, which is inconvenient for users. On the other hand, the passive localization is more practical but still not fully investigated with comprehensive tests.

2 The existing localization experiments are limited by the insufficient number of RPs and testing points per unit testing area as the measurements on the RPs are conducted manually. As the localization accuracy is determined by the density of the RPs in the target area, the performances of the reported algorithms are limited. Further, due to the limited number of measurements conducted manually, the previous works select fixed testing points or even on the same spots of the RPs and treat the localization as a classification model, which further departs the model generalization from practical cases.

3 The analysis of different working modes of the phones are mostly ignored. In practical scenarios, the phones behave differently due to the requirement of usage data, i.e., sending less WiFi frames in the idle mode and transmitting

more aggressively if users browse website or watch online video. Therefore, some existing passive indoor localization schemes, using data rate [88] or packet delivery ratio [15] fingerprints, are severely affected by those changes in real usage situations.

6.1.4 Contributions

To address the existing problems in literature, this chapter focuses on 2 most practical passive indoor localization scenarios including idle and transmission modes. In the idle scenario, the WiFi of the mobile device is on but the user is not using the phone at the moment (e.g., the user is moving and putting the phone in their pocket). Due to the limited sending frame in that mode, RTS/CTS mechanism is utilized to frequently obtain the RSSI data for localization process. On the other hand, in the transmission scenario, the mobile device is connected to an AP and sending data frames (e.g., the user is watching online videos or browsing the web). Both of RSSI and CSI fingerprints are available in this case.

The main contributions of this chapter are summarized as follows.

1 We present the first work which proposes comprehensive practical passive indoor localization schemes with detailed analysis of different work modes of the phone, i.e., idle, on-off screen, data transmission mode. Two most popular WiFi indoor localization scenarios, including idle and transmission modes along with RTS/CTS process utilization, are fully analyzed.

2 We propose several localization algorithms, which reasonably fit for each implemented practical scenario, to achieve the highest localization accuracy. In the idle scenario, SRL-KNN [8], P-MIMO LSTM [14] and SSP [59] are utilized to leverage RSSI data. On the other hand, in the transmission scenario, along

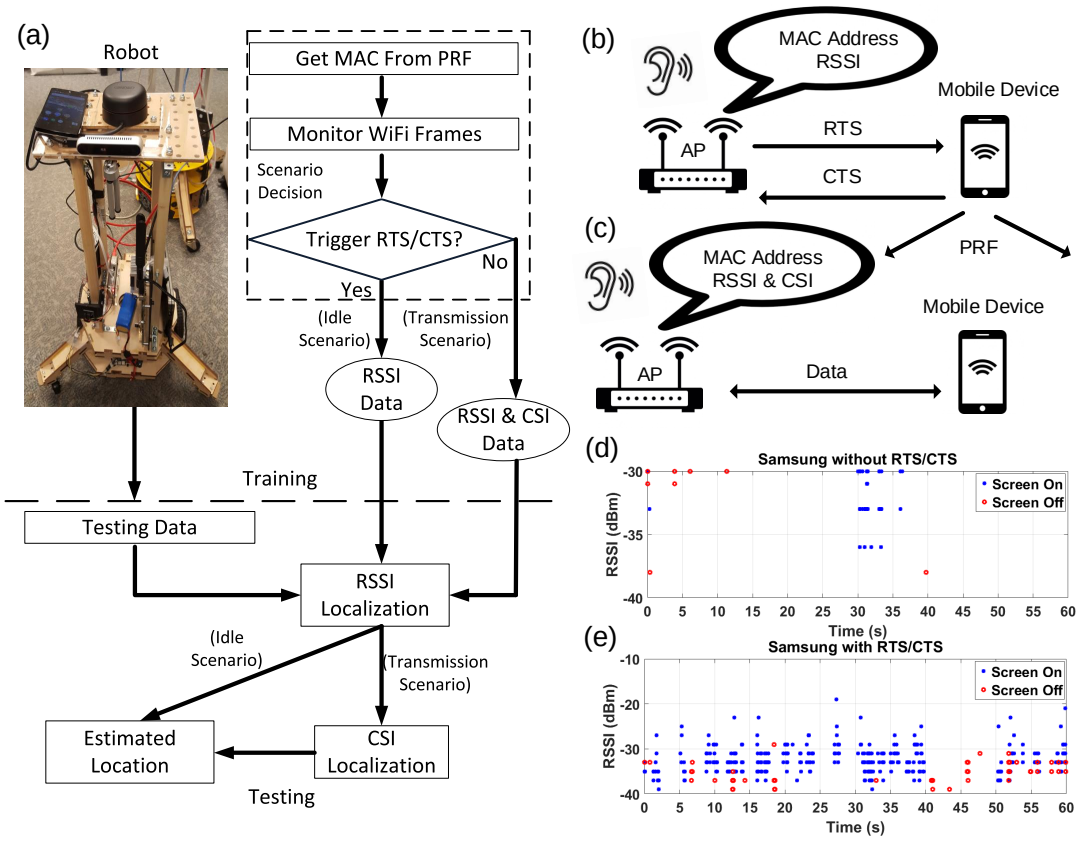


Figure 6.1: (a) Proposed Localization Process. (b) Idle Scenario. (c) Transmission Scenario. (d) Captured packets of Samsung Galaxy S6 without RTS/CTS process in idle scenario. (e) Captured packets of Samsung Galaxy S6 with RTS/CTS process in idle scenario.

with RSSI, additional CSI data correlation is utilized to enhance the accuracy.

3 The proposed models are tested in an extensive autonomous experiment with the support of the robot, which includes thousands of RPs, testing trajectories and a variety of phone types such as Samsung Galaxy S6, HTC OneX, Iphone X and LG Nexus 5.

6.2 Proposed Passive Localization System

6.2.1 Proposed Localization Process

As shown in Fig. 6.1(a), the proposed localization system is divided into two phases: a training phase and a testing phase. In the training phase, fingerprints at each predefined RP location are collected from all available APs and stored in a database. The fingerprints can be RSSI, CSI or both, depending on different usage scenarios. We assume that the area of interest has P APs and M RPs. For each RP i at its physical location $\boldsymbol{l}_i(x_i, y_i)$, a corresponding fingerprint vector is denoted as $\boldsymbol{F}(\boldsymbol{l}_i) = \{F_1(\boldsymbol{l}_i), F_2(\boldsymbol{l}_i), ..., F_N(\boldsymbol{l}_i)\}$, where N is the number of available features and $F_j(\boldsymbol{l}_i)$, $1 \leq j \leq N$, is the j-th feature at point i. In the testing phase, each unknown location of the user, denoted as a testing point, is determined by the localization algorithms, i.e, RSSI or CSI based approaches. Note here that the only assumption of our passive localization system is turning on WiFi of the mobile device.

Both training and testing phases start with the scenario decision steps. Firstly, when a user, carrying a mobile phone, walks to the interested area, all available P APs will monitor the probe request frames (PRF) sending from the phone. In the WiFi standard, a mobile device always periodically broadcasts PRF to discover existing 802.11 networks within its proximity [93]. Therefore, our AP monitor system will be guaranteed to capture at least one PRF if the WiFi of the phone is on. After PRF is successfully caught, media access control (MAC) address of the device will be retrieved. Then, the AP system continues to monitor all of the frames coming from that MAC address to decide whether the number of received WiFi frames are sufficient to determine user's location. Obviously, the more frames we receive, the better accuracy we achieve. In a practical scenario, the user location will be updated at a consecutive sampling time interval Δt (s). Therefore, the requirement is that

at least one WiFi frame is received every Δt. In order to consistently achieve good accuracy, we propose the localization solutions for 2 most popular practical modes of mobile devices including the idle scenario (Fig. 6.1(b)) and transmission scenario (Fig. 6.1(c)). The details of those 2 modes are described as follows.

6.2.2 Idle Scenario

In this scenario, a user, carrying a mobile phone, walks to the interested area without using it for any network purpose. The phone WiFi is on, but it might or might not connect to any specific AP. After acquiring the MAC address of this phone with PRF frame, our target is locating its position in every Δt. Due to the low data demanding, the mobile phone is currently in the idle mode which limits the number of sending WiFi frames to save the battery power. If the phone screen is locked or turned off, the number of WiFi frames will be even further reduced. Fig. 6.1(d) illustrates the timeline of sending frames from an idle phone, i.e., Samsung Galaxy S6, in a fixed location during 60 s. The frame intervals are marked with blue stars (on screen), red circles (off screen). The WiFi frames are monitored by Alfa AC1200 router in a 2.4 GHz channel. Without the interference of any special process from AP, the number of sending frames are very limited. In some window time, there is no received frame such as from second 15 to second 30, or from second 40 to second 60. Therefore, in those time periods, it is impossible to locate user's position. That is the key problem of passive indoor localization in general.

We propose to apply RTS/CTS process to solve the limitation of received packets in the idle scenario. RTS/CTS is a handshaking mechanism to reserve the channel for a specific duration before actual data transfer starts [94]. In Fig. 6.1(b), after having the MAC address of the targeted phone from PRF, every Δt, our corresponding AP will send K RTS frames to that MAC address. According to the IEEE 802.11 stan-

dard, while an AP transmits a RTS frame to a target, the target should immediately reply a CTS frame to the AP (even if the target is not associated with the AP) [15]. Therefore, the phone will be forced to respond several CTS frames back to the corresponding AP. Fig. 6.1(e) illustrates the timeline of sending frames from Samsung Galaxy S6 in the same 60 s period with RTS/CTS mechanism. The frequency of RTS from AP is 200 ms/frame. The number of sending frames from the phone increase significantly comparing with the previous case without RTS/CTS process. There is no long window time without any frame, which enables continuous localization ability. All of P available APs listen those frames with MAC filter to get RSSI information of the targeted phone. Several algorithms are then proposed in Subsection 6.2.4 to locate the user's position accurately based on these RSSI fingerprints.

6.2.3 Transmission Scenario

In this scenario (Fig. 6.1(c)), the user connects his/her mobile phone to a dedicated AP for network purposes (browsing websites or watching online videos). Multiple WiFi frames are sent from the phone, including data frame, ACK, CTS to maintain the connection. Therefore, RTS/CTS mechanism is not necessary to implement in this case. With the long preamble in data frame [81], not only RSSI but CSI can also be extracted from P APs. The localization accuracy in this scenario is expected to be higher than the idle scenario because more fingerprints can be achieved. The algorithm details with the combination of RSSI, CSI fingerprints will be presented in Subsection 6.2.4 and Subsection 6.4.2.

6.2.4 Localization Algorithms

There are different sets of localization algorithms which are proposed for each case. The target of the proposed approaches is getting good accuracy, while mitigating the

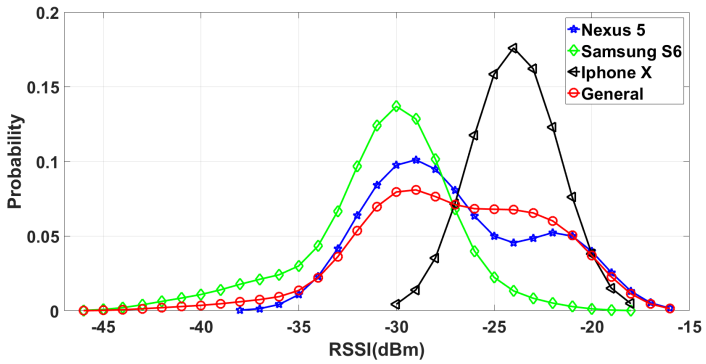


Figure 6.2: PDF of RSSI distributions between different phones and general PDF at a fixed location.

effects of device heterogeneity (mentioned in Subsection 6.1.1).

For Idle Scenario

In this scenario, the only available fingerprint is RSSI, which is relatively different in multiple mobile devices. Fig. 6.2 shows the RSSI PDF of 3 phone types, including Nexus 5, Samsung Galaxy S6 and Iphone X, monitored by an AP in 1 minute at a fixed location. The RSSI ranges in 3 phones are significantly different. The strongest transmission signal range belongs to Iphone X (-30 dBm to -18 dBm), following Nexus 5 (-38 dBm to -16 dBm) and Samsung S6 (-45 dBm to -18 dBm). Therefore, if we use the RSSI data from 1 phone, e.g., Nexus 5, for the training phase and test with another phone, e.g., Iphone X, the accuracy is deteriorated. The requirement here is utilizing all of RSSI data from multiple devices for the training.

Probabilistic methods with Kernel density estimator showing in [55,92] can represent well RSSI features over several devices. Therefore, we exploit that technique to estimate RSSI PDF along with semi-sequential SSP model [14] to achieve the localization accuracy in our scenario. The red circle line in Fig. 6.2 illustrates the general RSSI PDF out of 3 phones data using Kernel density estimator. The general PDF covers the whole area of all phones (-45 dBm to -16 dBm). After the PDF is estimated

in the training phase with Kernel method, the location of the user will be determined based on SSP model [14] as

$$P(\boldsymbol{l}_{curr} \approx \boldsymbol{l}_i | \boldsymbol{F}(\boldsymbol{l}_{curr}), \boldsymbol{l}'_{pre}) \propto \prod_{k=1}^{N} P(F_k(\boldsymbol{l}_{curr}) | \boldsymbol{l}_i) P(\boldsymbol{l}_i | \boldsymbol{l}'_{pre}). \tag{6.1}$$

In the formula, $\boldsymbol{F}(\boldsymbol{l}_{curr})$ is the fingerprint vector of the unknown current location $\boldsymbol{l}_{curr}$, while $\boldsymbol{l}'_{pre}(x'_{pre}, y'_{pre})$ is the predicted previous location. $P(\boldsymbol{l}_i | \boldsymbol{l}'_{pre})$ is soft range SSP window. The likelihood function $P(F_k(\boldsymbol{l}_{curr}) | \boldsymbol{l}_i)$ describes the probability of the k-th fingerprint feature at location $\boldsymbol{l}_i$.

On the other hand, neural network and especially RNN which exploits the additional information in time domain proves its effectiveness in solving the localization in multiple devices [59]. Instead of depending on one instant RSSI scan, RNN network aims at trajectory positioning and takes into account the correlation among the RSSI measurements in a trajectory. Therefore, it partially mitigate the device-dependant problem of RSSI. Among all of the existing network structure, P-MIMO LSTM [59] provides the best accuracy with the robustness in multiple databases and environments. Therefore, in this chapter, we choose to implement P-MIMO LSTM for our experiment.

For Transmission Scenario

In this scenario, there are more available fingerprint information, including RSSI and especially CSI. CSI amplitudes can be grouped into a time-frequency matrix that resembles CSI images [5, 95]. These images are the fingerprint features for each location. On the other hand, raw CSI phase is prone to noise and random fading [61]. Therefore, the preprocessing step is needed before using it as a feature [5]. Several researchers use the phase difference between subcarriers instead of raw phase as the

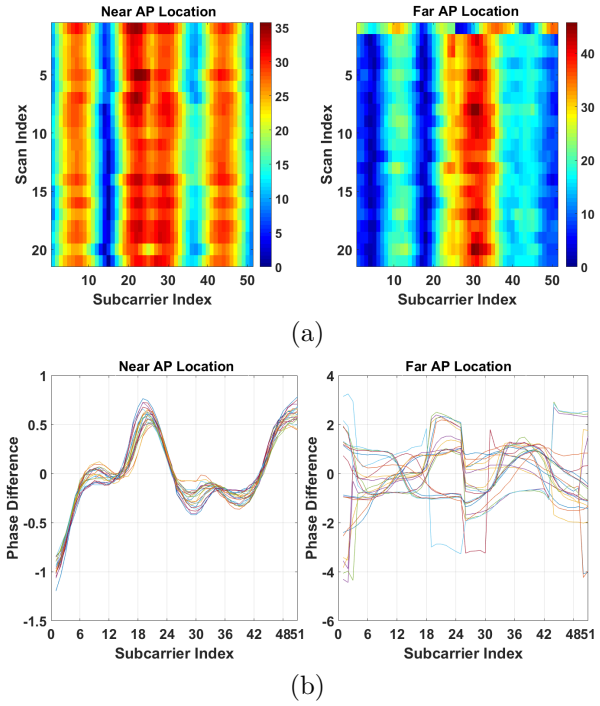


Figure 6.3: CSI fingerprint features of Samsung Galaxy S6 at 2 fixed locations (a) CSI amplitude images. (b) CSI phase difference.

fingerprint like in CIFI [77].

Fig. 6.3 illustrates an example of CSI fingerprint features of Samsung Galaxy S6 at 2 fixed locations, i.e., one is near AP (within 1 m) and has the line of sight (LOS), another one is further away (more than 10 m) and does not have LOS (NLOS). Fig. 6.3(a) presents the CSI amplitude image constructed by the method in [95], while Fig. 6.3(b) shows the corresponding phase difference information following the method in CIFI [77]. There are 20 received scans (packets) in each case. Clearly, CSI information at the near AP location is more stable than the further ones. The color of CSI image at near AP location is more solid than the far AP location. Furthermore, CSI phase information of the close location follows the same trend in different scans, while far AP location phase is totally different between each scan. The reason is that CSI is sensitive to the change of environments especially the interference of human blocking and movements [95]. The larger the area is, the more chance the environment can

change (people or some moving objects can interfere in the middle). Therefore, using only CSI fingerprints for localization in the big area might lead to low accuracy, i.e., 2.5 m in [95].

Our proposed method for this transmission scenario (Fig. 6.1(a)) firstly starts with RSSI localization (SSP or P-MIMO LSTM) as explained in Subsection 6.2.3. Instead of getting the estimated location directly, RSSI information helps to determine the possible area of the user by choosing K nearest neighbour locations. After having K candidate locations, CSI information will be utilized to select the best suitable point among those K ones. CSI amplitude and phase of the testing point is correlated with the CSI information of the K neighbour locations using Pearson coefficient [8]. The location which provides the largest coefficient will become the final output location. More details of the results will be presented in Subsection 6.4.2.

6.3 Database And Experiments

All experiments have been carried out in 2 different environments, including office and home. The office experiment (Fig. 6.4(a)) is conducted on the third floor of Engineering Office Wing (EOW), University of Victoria, BC, Canada. The dimension of the area is 21 m $\times$ 16 m. Besides, the home experiment (Fig. 6.4(b)) is in the basement of 3850 Cardie Court, Victoria, BC, Canada with the the area being 15 m $\times$ 8 m. The fingerprints for both training and testing are collected using an autonomous driving robot. The 3-wheel robot (Fig. 6.1(a)) has multiple sensors including a wheel odometer, an inertial measurement unit (IMU), a LIDAR, 3 sonar sensors and a color and depth (RGB-D) camera. It can navigate to a target location within an accuracy of 0.21$\pm$0.02 m.

During the experiments, the robot carried the mobile devices including Samsung

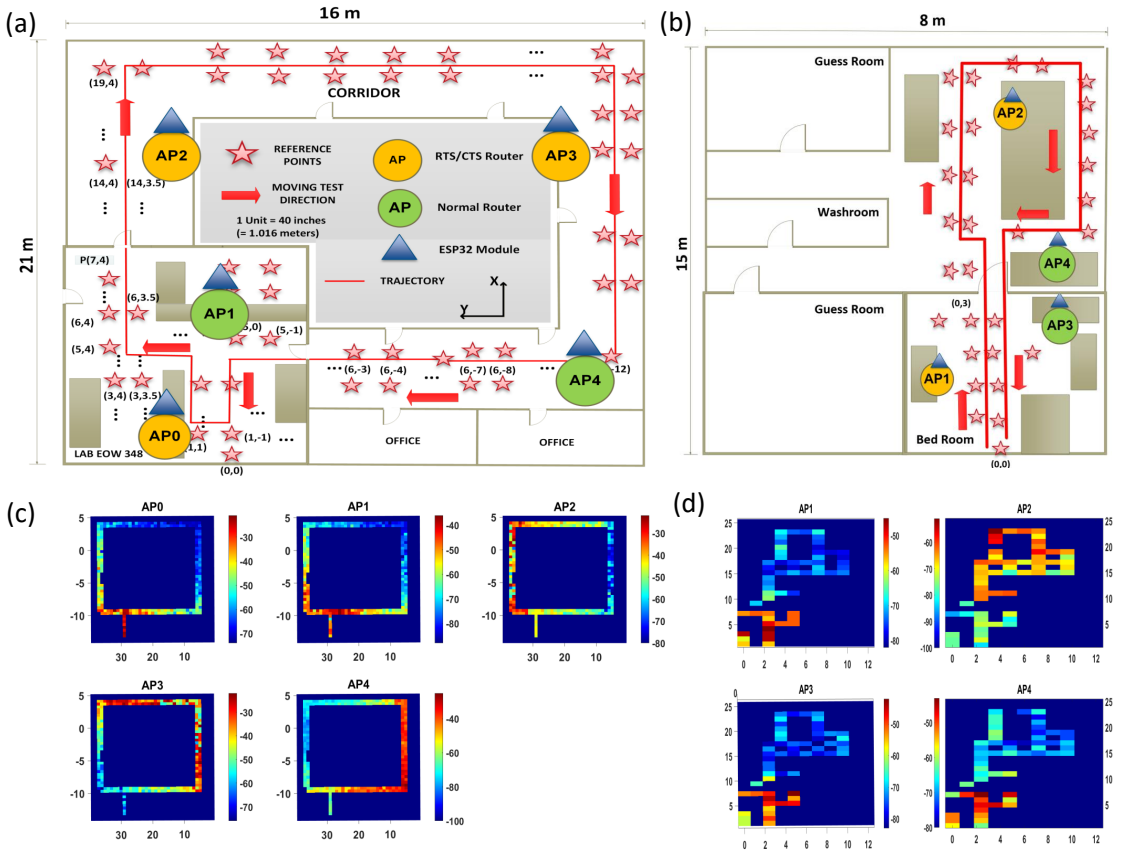


Figure 6.4: (a) Office experiment floor map. (b) Home experiment floor map. (c) RSSI heat map of office experiment. (d) RSSI heat map of home experiment.

Galaxy S6, HTC OneX, Iphone X and LG Nexus 5 to collect the WiFi fingerprints. In the training phase, the robot stays at every RP to collect data for 2 minutes. The data is repeatedly collected in different days and at different time to build the database in both scenarios as explained in Section 6.2. In the testing phase, the robot navigates along a pre-defined route (Fig. 6.4(a), (b)) at a speed randomly varying within 0.6-4.0 m/s to simulate the indoor walking pattern of a normal person [41]. The test experiment is conducted several rounds per day and repeated in different days.

Regarding the office experiment, there are 5 APs with 3 of them being installed libtins library [96] for crafting the WiFi packets to trigger RTS/CTS process. In the home experiment, since the area is smaller, there are 4 APs with 2 of them being able

to run RTS/CTS mechanism. Those APs include Alfa AC1200, TPlink AC1750 and Intel 5300 NIC. Fig. 6.4(c), (d) illustrate the heat map of the RSSI collected from the a smartphone in both environments, where the signal strength is represented by color. Because the transmitting power of the phones might vary due to different working modes, some parts in the RSSI heat map are missed or not consistent. However, the signals are still clearly stronger in the area near the AP. The chosen AP sets can cover the whole area in both cases with total 1600 RPs (pink stars) and 900 testing points (solid red line). For the transmission scenario, ESP32-S2 modules are attached with all of the normal routers. ESP32 is a low-cost, low-power system on a chip microcontroller with integrated WiFi. It supports to get CSI information from WiFi data packets [97]. Fig. 6.3 illustrates the example of CSI amplitude and phase collected from ESP32 at the bandwidth 20 MHz, frequency 2.4 GHz. There are total 51 useful subcarriers extracted from the WiFi packets.

Note that the validation data is collected at different time from the training data but at the same RP locations, while testing point locations are randomly selected to be different from all RPs in order to approximate our model closer to practical situations. The user location is updated every Δt =1-2 s. Therefore, the sending frequency of crafting RTS frames to the phone is chosen as 200 ms per frame to guarantee that every Δt we have at least few RSSI scans for the localization.

6.4 Results And Discussions

6.4.1 Idle Scenario

As explained in Subsection 6.2.4, there are 2 proposed algorithms including probabilistic model SSP [14] and neural network model P-MIMO LSTM [59] to achieve good accuracy, while mitigating the effects of device heterogeneity. For SSP model,

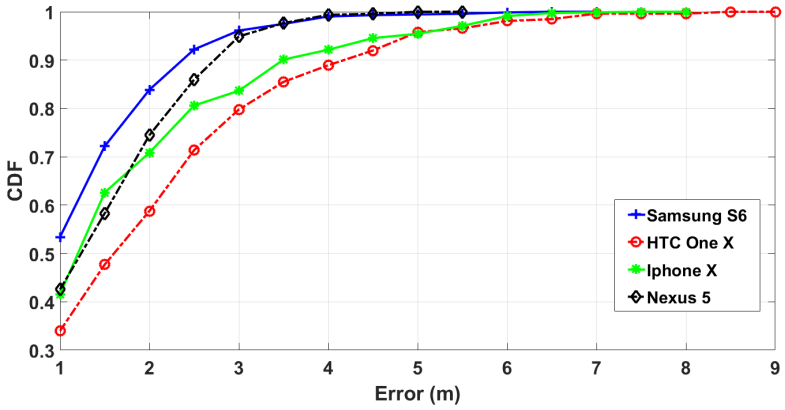


Figure 6.5: CDF of the localization error of the proposed SSP model with different phones.

Kernel method [46] and Gaussian window are used since they provide the best accuracy among others according to the experiments in [14]. As the robot maximum speed, v_{max} is pre-configured to be 4 m/s, the maximum distance a user can travel during consecutive measurements is $d_{max} = v_{max}\Delta t$ =4 m. Therefore, we use a Gaussian window with the spread $\sigma = d_{max}$. The probability distribution of RSSI is constructed by using the kernel smoothing function in the database of 20,000 collected scans from all of the phones. Fig. 6.5 compares the cumulative distribution function (CDF) of localization errors of SSP models with different phones including Samsung Galaxy S6, HTC OneX, Iphone X and LG Nexus 5. Samsung S6 provides the best accuracy with 80% of the localization errors smaller than 2 m, with more than 50% of testing location errors less than 1 m. On the other hand, Nexus 5 and Iphone X have 80% less than 2.5 m, while 80% of HTC One X errors are below 3 m. Besides, the maximum error of HTC One X is 9 m, which is nearly 2 times higher than that of Samsung S6. Table 6.4 illustrates the average error comparison of SPP models between those 4 phones. Al of the phones have the average localization error less than 2 m with the best accuracy is achieved by Samsung S6, i.e., 1.2 ± 0.9 m.

For P-MIMO LSTM model, the setup parameters follows the optimal set proposed

Table 6.2: Setup parameters for P-MIMO LSTM

Category	**Value**
RNN type	LSTM
Memory length (T)	10
Model	P-MIMO
Loss function	RMSE
Hidden layer (HL)	2
Number of neurons for each HL	100
Dropout	0.2
Optimizer	Adam
Learning rate	0.001
Number of training trajectory	110,000
Number of validation trajectory	29,700
σ	4 m
Δ_t	1 s
d_{max}	4 m

Table 6.3: Average Localization Errors of Proposed Models (meter)

Method	**HTC**	**Samsung**	**Nexus 5**	**Iphone X**
SSP Model	1.9 ± 1.5	1.2 ± 0.9	1.4 ± 0.9	1.6 ± 1.4
P-MIMO LSTM	1.8 ± 1.5	1.2 ± 1.0	1.4 ± 1.2	1.2 ± 0.9

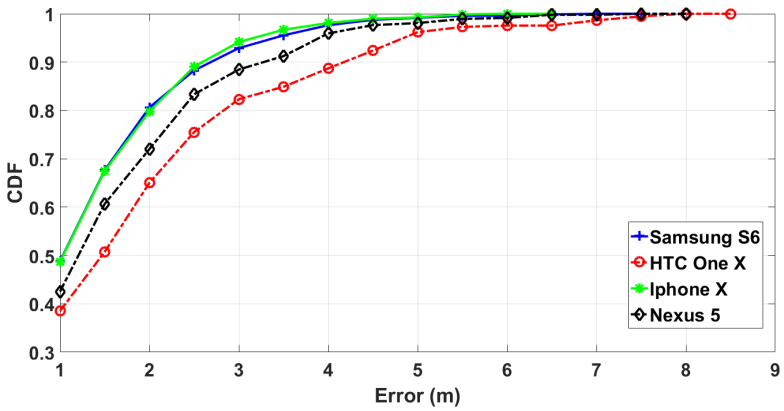


Figure 6.6: CDF of the localization error of P-MIMO LSTM model with different phones.

in [59], which is also summarized in Table 6.2. Fig. 6.6 compares the CDF of the localization error of P-MIMO LSTM model. Samsung S6 and Iphone X equivalently have the best performance with 80% localization errors being below 2 m and maximum errors are 6.5 m and 7.5 m, respectively. HTC has 80% of errors below 3 m and maximum errors can reach to 8.5 m.

In general, Table 6.4 indicates that P-MIMO LSTM has slightly better perfor-

Table 6.4: Localization Errors Comparison Between Scenarios (meter)

Method	**HTC**	**Samsung**	**Nexus 5**	**Iphone X**
SSP (Idle)	1.9 ± 1.5	1.2 ± 0.9	1.4 ± 0.9	1.6 ± 1.4
SSP (Transmission)	1.2 ± 0.9	1.0 ± 0.7	1.2 ± 0.9	1.0 ± 0.8

mance than SSP model. However, SSP model does not require the training phase, which compares the current RSSI measurement with the ones in the database to get user locations directly. In contrast, P-MIMO LSTM requires to train the neural network model beforehand. In the experiment, the training is executed on a home-built computer with an AMD FX(tm)-8120 Eight-Core CPU and an Nvidia GTX 1050 GPU. The running time is approximately 4 s per epoch. Therefore, the training time is approximately $\cong$ 1 hour and 6 minutes.

6.4.2 Transmission Scenario

As explained in Subsection 6.2.3, the localization process started with RSSI localization algorithm to pick K possible candidates. The chosen algorithm here is SSP and $K = 5$. Table 6.4 illustrates the average error comparisons between 2 scenarios including idle and transmission. In general, with the additional information from CSI, the accuracy of localization enhances approximately 20% - 40%. HTC and Iphone X have the biggest improvement with the average error decreasing from 1.9 ± 1.5 m and 1.6 ± 1.4 m in idle scenario to 1.2 ± 0.9 m and 1.0 ± 0.8 m in transmission scenario, respectively. On the other hand, Samsung and Nexus 5 witness a slighter enhancement with the decrease of 0.2 m comparing between 2 scenarios.

Fig. 6.7 illustrates the CDF errors of 4 phones. In this scenario, the performance of those phones is getting closer together than the transmission scenario. 80% average errors of HTC and Nexus 5 are all below 2.5 m, while the ones of Samsung and Iphone X are below 1.8 m. The maximum errors of HTC is 5.5 m, whereas those of the other

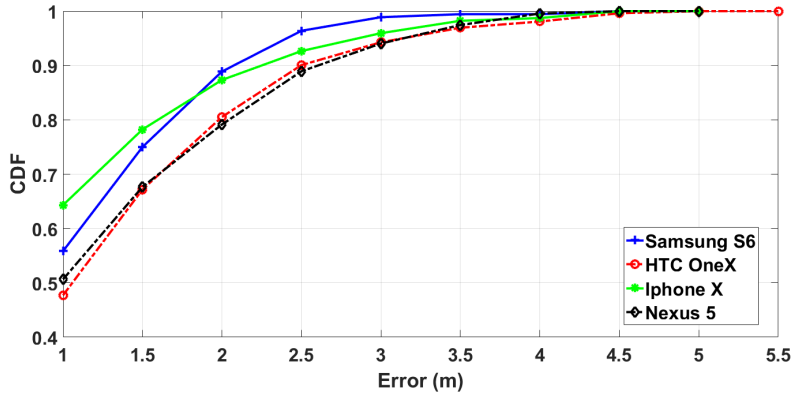


Figure 6.7: CDF of the localization error of SSP model with different phones.

phones are around 5 m.

6.5 Conclusion

In conclusion, we have proposed a comprehensive practical solution to leverage the received data at the routers to locate the position of a mobile user. In this passive indoor localization scenario, the problem of data insufficiency is mitigated by RTS/CTS mechanism. Two most popular mobile device usage scenarios, i.e., idle and transmission modes, are analyzed in details. The completed algorithms for both cases proposed are SSP or P-MIMO LSTM for the idle scenario and an additional Pearson correlation step for the transmission scenario. The localization accuracy has been investigated through extensive on-site experiments using autonomous robots with several phones, e.g., Nexus 5, Samsung, Iphone and HTC, in hundreds of testing locations. The baseline accuracy is 1.5 m and 1.0 m for idle and transmission scenario respectively.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation, we have studied autonomous indoor localization solutions using WiFi signals. Both active and passive localization models with RSSI and CSI fingerprints are analysed in details. Our proposed methods are practical and able to locate a walking human who carries a mobile device with few feet accuracy. The following outlines the contributions we have achieved.

- In Chapter 2, for a low complexity application, the proposed SRL-KNN algorithm exploits the information of previous positions and simultaneously applies the soft range limiting factor for fingerprint distance calculation to achieve more accurate and stable positioning performance. We demonstrated that SRL-KNN can address effectively some main challenges of KNN including the spatial ambiguity, RSSI instability and the RSSI short collecting time, especially when RSSI histogram is taken into account in calculating fingerprint distance. Experimental results have shown that SRL-KNN achieves the best accuracy of 0.66 m with 80% of the error within 0.89 m, which outperforms existing KNN methods.

- In Chapter 3, a simple but efficient semi-sequential probabilistic model applies an additional short term memory step to enhance the performances of the indoor localization probabilistic approaches. The proposed SSP model is flexible and can be implemented to most of the conventional methods. This model leverages the information of the previous position to effectively determine the candidate location probability since the user's speed in an indoor environment is bounded. Three soft range windows including Gaussian, Hann and Tukey have been proposed as SSP models. Several experiments have been conducted including both RSSI and CSI fingerprints to demonstrate that SSP reduces the maximum error and boosts the performance of other existing probabilistic approaches by at least $25\% - 30\%$.

- In Chapter 4, taking into account the advantage of neural network to solve high dimension and highly nonlinear problem, a comprehensive RNN solution for WiFi RSSI fingerprinting with detailed analysis and comparisons is proposed. The relation between a series of the RSSI measurements is utilized to determine the user's moving path as one problem. Experimental results have consistently demonstrated that our LSTM structure achieves an average localization error of 0.75 m with 80% of the errors under 1 m, which outperforms feed-forward neural network, conventional methods such as KNN, Kalman filter and probabilistic methods. Furthermore, main challenges of those conventional methods including the spatial ambiguity, RSSI instability and the RSSI short collecting time have been effectively mitigated. Besides, the analysis of vanilla RNN, LSTM, GRU and BiLSTM with important parameters have been discussed in details, i.e, loss function, memory length, input and output features, etc.

- In Chapter 5, we explore the algorithm to locate a mobile device in the environment which has only a single available WiFi AP. The proposed CNN-LSTM

network extracts both spatial and temporal information of received CSI signals to determine a user's moving path. The extensive on-site experiments, with several mobile devices including mobile phone (Nexus 5) and laptop (Intel 5300 NIC) in hundreds of testing locations, have been presented. The experimental results have consistently demonstrated that our CNN-LSTM structure achieves an average localization error of 2.5 m with 80% of the errors under 4 m, which outperforms the other algorithms by at least 50% under the same test environment.

- In Chapter 6, in order to enhance the practicality, this chapter focuses on the passive WiFi indoor localization system. Most of the conventional methods rely on the collected WiFi signal on the mobile devices (active information), which requires a dedicated software to be installed. Different from them, we leverage the received data of the routers (passive information) to locate the position of the user. The problem of data insufficiency in passive indoor localization is mitigated by RTS/CTS mechanism. Furthermore, the completed localization solutions for two most popular mobile device usage scenarios, i,e., idle and transmission modes, are analyzed in details. Two kinds of fingerprints, including RSSI and CSI, along with several types of algorithms, including deterministic, probabilistic and neural networks are utilized to get the baseline accuracy for each scenario. The localization accuracy is investigated through extensive on-site experiments with several phones, e.g., Nexus 5, Samsung, Iphone and HTC, in hundreds of testing locations. The experimental results demonstrate that our proposed localization scheme achieves an average localization error of around 1.5 m when the phone is in idle mode, and approximately 1 m when it actively transmits data.

7.2 Future Work

For the future work that plans beyond this dissertation, there are still many open issues. Some important points can be further discussed are listed below.

- Although SRL-KNN in Chapter 2 and SSP in Chapter 3 provides good accuracy in general indoor localization environment, they have the limited performance in fast speed variation scenario. As explained in Chapter 4, because of the short memory, these methods have the constraints on the speed of the users. If a user changes moving speed rapidly, the localization accuracy of them will be degraded severely. The solution to extend the memory length to several more steps and maintain the low complexity needs to be further investigation. Furthermore, in SSP model, the condition of mutual independent relationship between the fingerprints of different routers might not hold among the set of routers. In that case, all of the probabilities proposed in SSP are approximations and the localization accuracy needs to be re-estimated.

- Compared with other conventional methods, our proposed P-MIMO LSTM model in Chapter 4 outperforms. However, those conventional methods do not require the training phase, which compares the current RSSI measurement with the ones in the database to get user locations directly. In contrast, the neural network methods require to train the model beforehand. In practical scenarios, more APs are used, more RSSI features can be extracted and better performance can be achieved. However, increasing the number of APs also creates more computational cost and extends the training time. In the future work, when the number of APs are increased to achieve better accuracy, the solution to reduce the training time of our P-MIMO LSTM model needs to be further discussion.

- As mentioned in Chapter 5, although CSI provides rich information from multiple subcarriers, it is sensitive to the change of environments especially the interference of human blocking and movements. Therefore, our proposed CNN-LSTM structure only achieves a limited accuracy of around 2.5 m. In the future research, the way to adopt more kinds of fingerprints including both RSSI and CSI to improve the performance can be further investigated.

- In Chapter 6, although RTS/CTS mechanism is practical and can be applied to every phone, it also interferes with WiFi channel and might affect to the data transmission of other devices. Therefore, the sending CTS frequency of the router beckons further investigation. On the other hand, in real scenario with different number of routers, how to coordinate the routers and synchronize them together to get the best performance is also worthy for more discussion.

Chapter 8

Publications

1. **Minh Tu Hoang**, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate RSSI indoor localization," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10639 - 10651, Dec. 2019.

2. **Minh Tu Hoang**, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy Tarimala, " Semi-sequential probabilistic model for indoor localization enhancement, ", IEEE Sensors Journal, vol. 20, no. 11, pp. 6160 - 6169, Jun. 2020.

3. **Minh Tu Hoang**, Y. Zhu, B. Yuen, T. Reese, X. Dong, T. Lu, R. Westendorp, and M. Xie, "A Soft Range Limited K-Nearest Neighbours Algorithm for Indoor Localization Enhancement," IEEE Sensor Journal, vol. 18, pp.10208 – 10216, Dec. 2018.

4. **Minh Tu Hoang**, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "A CNN-LSTM quantifier for single access point CSI indoor localization," arxiv, pp. 111, Apr. 2020.

Bibliography

[1] Suining He and S.-H. Gary Chan. Wi-Fi fingerprint-based indoor positioning:recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(11):466–490, February 2016.

[2] Yaqian Xu. *Autonomous Indoor Localization Using Unsupervised Wi-Fi Fingerprinting*. Kassel University Press, 2016.

[3] C.Gentile, N. Alsindi, R.Raulefs, and C.Teolis. *Geolocation Techniques Principles and Applications*. Springer, 2013.

[4] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 37(6):1067 – 1080, November 2007.

[5] Hao Chen, Yifan Zhang, Wei Li, Xiaofeng Tao, and Ping Zhang. ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information. *IEEE Access*, 5:18066 – 18074, September 2017.

[6] Xuyu Wang, Xiangyu Wang, and Shiwen Mao. CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi. *IEEE International Conference on Communications (ICC)*, pages 1938–1883, July 2017.

[7] Xuyu Wang, Lingjun Gao, and Shiwen Mao. BiLoc: Bi-modal deep learning for indoor localization with commodity 5 GHz WiFi. *IEEE Access*, 5:4209 – 4220, March 2017.

[8] Minh Tu Hoang, Yizhou Zhu, Brosnan Yuen, Tyler Reese, Xiaodai Dong, Tao Lu, Robert Westendorp, and Michael Xie. A soft range limited K-nearest neighbours algorithm for indoor localization enhancement. *IEEE Sensors Journal*, 18:10208–10216, December 2018.

[9] Junghyun Jun, Liang He, Yu Gu, Wenchao Jiang, Gaurav Kushwaha, Vipin A, Long Cheng, Cong Liu, and Ting Zhu. Low-overhead WiFi fingerprinting. *IEEE Transactions on Mobile Computing*, 17:590 – 603, 2018.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9:1735–1780, November 1997.

[11] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoderdecoder approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, page 103111, October 2014.

[12] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673 – 2681, November 1997.

[13] Wanlong Zhao, Shuai Han, Rose Qingyang Hu, Weixiao Meng, and Ziqing Jia. Crowdsourcing and multi-source fusion based fingerprint sensing in smartphone localization. *IEEE Sensors Journal*, 18:3236 – 3247, February 2018.

[14] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy. Semi-sequential probabilistic model for indoor localization enhancement. *IEEE Sensors Journal, Early Access*, 20(11):6160–6169, June 2020.

[15] Y. Duan, K. Lam, V. C. S. Lee, W. Nie, H. Li, and J. K. Ng. Packet delivery ratio fingerprinting: Toward device-invariant passive indoor localization. *IEEE Internet of Things Journal*, 7(4):2877–2889, 2020.

[16] G. Deak, K. Curran, and J. Condell. A survey of active and passive indoor localisation systems. *Computer Communications*, 35:1939 – 1954, 2012.

[17] Chen Liu, Dingyi Fang, Zhe Yang, Hongbo Jiang, Xiaojiang Chen, Wei Wang, Tianzhang Xing, and Lin Cai. RSS distribution-based passive localization and its application in sensor networks. *IEEE Transactions on Wireless Communications*, 15:2883 – 2895, April 2016.

[18] M. Brunato and R. Battiti. Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks*, 47:825–845, April 2005.

[19] Shih-Hau Fang and Tsung-Nan Lin. Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments. *IEEE Transactions on Neural Networks*, pages 1973–1978, Nov 2008.

[20] Ke Shi, Z. Ma, R. Zhang, W. Hu, and H. Chen. Support vector regression based indoor location in IEEE 802.11 environments. *Mobile Information Systems*, 2015:1–14, August 2015.

[21] P. Bahl and V.N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. *IEEE Infocom*, pages 775–784, March 2000.

[22] Yaqin Xie, Yan Wang, Arumugam Nallanathan, and Lina Wang. An improved K-Nearest-Neighbor indoor localization method based on Spearman distance. *IEEE Signal Processing Letters*, 23:351 – 355, 2016.

[23] M. Youssef and A. Agrawala. The Horus WLAN location determination system. *Proc. 3rd international conference on Mobile systems, applications, and services*, pages 205–218, 2005.

[24] Anthea Wain Sy Au, Chen Feng, Shahrokh Valaee, Sophia Reyes, Sameh Sorour, Samuel N. Markowitz, Deborah Gold, Keith Gordon, and Moshe Eizenman. Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device. *IEEE Transactions on Mobile Computing*, 12:2050–2062, October 2013.

[25] I. Guvenc, C.T. Abdallah, R. Jordan, and O. Dedeoglu. Enhancements to RSS based indoor tracking systems using Kalman filters. *Proc. Global Signal Processing Expo and Intl Signal Processing Conf*, pages 1–6, 2003.

[26] A. Besada, A.M. Bernardos, P. Tarrio, and J.R. Casar. Analysis of tracking methods for wireless indoor localization. *Proc. Second Intl Symp. Wireless Pervasive Computing*, pages 493–497, Feb 2007.

[27] A. Kushki, K.N. Plataniotis, and A.N. Venetsanopoulos. Location tracking in wireless local area networks with adaptive radio maps. *Proc. IEEE Intl Conf. Acoustics, Speech and Signal Processing*, 5:741–744, May 2006.

[28] Yogita Chapre, Prasant Mohapatray, Sanjay Jha, and Aruna Seneviratne. Received signal strength indicator and its analysis in a typical WLAN system. *IEEE Conference on Local Computer Networks*, pages 1–4, 2013.

[29] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50:425–436, 02 2002.

[30] F. Evennou and F. Marx. Improving positioning capabilities for indoor environments with WiFi. *Proc. IST Summit*, pages 1–5, 2005.

[31] Jung Min Pak, Choon Ki Ahn, Peng Shi, Yuriy S. Shmaliy, and Myo Taeg Lim. Distributed hybrid Particle/FIR filtering for mitigating NLOS effects in TOA-based localization using wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 64:5182 – 5191, June 2017.

[32] Ran Liu, Chau Yuen, Tri-Nhut Do, and U-Xuan Tan. Fusing similarity-based sequence and dead reckoning for indoor positioning without training. *IEEE Sensors Journal*, 17(13):4197–4207, July 2017.

[33] C Wu, J Xu, Z Yang, N Lane, and Z Yin. Gain without pain: Accurate WiFi - based localization using fingerprint spatial gradient. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1:1–19, 2017.

[34] Bang Wang, Qiuyun Chen, LaurenCe T. Yang, and Han-Chieh Chao. Indoor smartphone localization via fingerprint crowdsourcing: challenges and approaches. *IEEE Wireless Communications*, 3:82–89, June 2016.

[35] Han Zou, Ming Jin, Hao Jiang, Lihua Xie, and Costas J. Spanos. WinIPS: WiFi-based non-intrusive indoor positioning system with online radio map construction and adaptation. *IEEE Transactions on Wireless Communications*, 16:8118–8130, Dec 2017.

[36] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40:3351, 1975.

[37] Beomju Shin and Jung Ho Lee. Enhanced weighted K-Nearest Neighbor algorithm for indoor Wi-Fi positioning systems. *International Journal of Networked Computing and Advanced Information Management*, 2(2):1–4, 2012.

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

[39] Bulut Altintas and Tacha Serif. Indoor location detection with a RSS-based short term memory technique (KNN-STM). *Eighth IEEE PerCom Workshop on Pervasive Wireless Networking*, pages 794–798, 2012.

[40] Raymond C. Browning, Emily A. Baker, Jessica A. Herron, and Rodger Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of Applied Physiology*, 100:390–398, February 2006.

[41] Betty J. Mohler, William B. Thompson, Sarah H. Creem-Regehr, Herbert L. PickJr, and William H. Warren. Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research*, 181:221–228, August 2007.

[42] Kamol Kaemarungsi and Prashant Krishnamurthy. Analysis of WLANs received signal strength indication for indoor location fingerprinting. *Pervasive Mobile Computing*, 8:292–316, April 2012.

[43] F. Dong, Y. Chen, J. Liu, and S. Piao. A calibration-free localization solution for handling signal strength variance. *Proc. 2nd international conference on Mobile entity localization and tracking in GPS-less environments,*, pages 79–90, 2009.

[44] Xiaohua Tian, Mei Wang, Wenxin Li, Binyao Jiang, Dong Xu, Xinbing Wang, and Jun Xu. Improve accuracy of fingerprinting localization with temporal correlation of the RSS. *IEEE Transactions on Mobile Computing*, 17:113–126, Jan 2018.

[45] Lei Yang, Hao Chen, and Yifan Zhang. Probabilistic-KNN: A novel algorithm for passive indoor-localization scenario. *Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2015.

[46] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Kernel-based positioning in wireless local area networks. *IEEE Trans. Mobile Comput.*, 6:689–705, June 2007.

[47] Chenshu Wu, Zheng Yang, and Chaowei Xiao. Automatic radio map adaptation for indoor localization using smartphones. *IEEE Transaction on Mobile Computing*, 17:517 – 528, March 2018.

[48] J. Torres-Sospedra and etc. Ujiindoorloc: A new multi-building and multi floor database for WLAN fingerprintbased indoor localization problem. *Indoor Positioning and Indoor Navigation*, pages 1–10, October 2014.

[49] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, thirdquarter 2019.

[50] K. Kaemarungsi and Krishnamurthy. Properties of indoor Received Signal Strength for WLAN location fingerprinting. *In Proceedings of IEEE First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 11–23, August 2004.

[51] L. Chen, B. Li, and Z. Zheng. An improved algorithm to generate a Wi-Fi fingerprint database for indoor positioning. *Sensors*, 13:11085–11096, 2013.

[52] Ville Honkavirta, Tommi Perala, Simo Aliloytty, and Robert Piche. A comparative survey of WLAN location fingerprinting methods. *6th Workshop on Positioning, Navigation and Communication*, pages 243–251, March 2009.

[53] Qideng Jiang, Yongtao Ma, Kaihua Liu, and Zhi Dou. A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization. *IEEE Sensors Journal*, 16:3764 – 3774, 2016.

[54] Teemu Roos, Petri Myllymki, Henry TirriPauli, and MisikangasJuha Sievnen. A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, 9:155–164, July 2002.

[55] C. Figuera, I. Mora, and A.G. Curieses. Nonparametric model comparison and uncertainty evaluation for signal strength indoor location. *IEEE Trans. Mobile Comput.*, 8:1250–1264, 2009.

[56] Wei Sun, Min Xue, Hongshan Yu, Hongwei Tang, and Anping Lin. Augmentation of fingerprints for indoor WiFi localization based on gaussian process regression. *IEEE Transactions on Vehicular Technology*, 67:10896 – 10905, November 2018.

[57] Simon Yiu and Kai Yang. Gaussian process assisted fingerprinting localization. *IEEE Internet of Things Journal*, 3:683 – 690, October 2016.

[58] Kaishun Wu, Jiang Xiao, Youwen Yi, Dihu Chen, Xiaonan Luo, and Lionel M. Ni. CSI - based indoor localization. *IEEE Transactions on Parallel and Distributed Systems*, 24:1300 – 1309, July 2013.

[59] Minh Tu Hoang, Brosnan Yuen, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy. Recurrent neural networks for accurate RSSI indoor localization. *IEEE Internet of Things Journal (Minor Revision)*, 2019.

[60] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM Comput. Commun. Rev.*, 41(1):53, January 2011.

[61] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66:763 – 776, January 2017.

[62] Roberto Battiti, Alessandro Villani, and Thang Le Nhat. Neural network models for intelligent networks: deriving the location from signal patterns. *in: Proceedings of AINS2002, UCLA*, pages 1–13, 2002.

[63] Xiaoxuan Lu, Han Zou, Hongming Zhou, Lihua Xie, and Guang-Bin Huang. Robust extreme learning machine with its application to indoor positioning. *IEEE Transaction on Cybernetics*, 46(1):194 – 205, January 2016.

[64] Huan Dai, Wen hao Ying, and Jiang Xu. Multi-layer neural network for received signal strength-based indoor localization. *IET Communications*, 10:717 – 723, January 2016.

[65] Jichao Jiao, Fei Li, Zhongliang Deng, and Wenjing Ma. A smartphone camera-based indoor positioning algorithm of crowded scenarios with the assistance of deep CNN. *Sensors (Basel)*, 17:1–21, March 2017.

[66] Ho Jun Jang, Jae Min Shin, and Lynn Choi. Geomagnetic field based indoor localization using recurrent neural networks. *GLOBECOM 2017*, pages 1–6, December 2017.

[67] G. Huang, S. Song, C. Wu, and K. You. Robust support vector regression for uncertain input and output data. *IEEE Trans. Neural Netw. Learn. Syst*, 23:1690–1700, November 2012.

[68] Yuan Lukito and Antonius Rachmat Chrismanto. Recurrent neural networks model for WiFi -based indoor positioning system. *International Conference on*

Smart Cities, Automation & Intelligent Computing Systems, Indonesia, pages 1–5, November 2017.

[69] Xuyu Wang, Zhitao Yu, and Shiwen Mao. DeepML: Deep LSTM for indoor localization with smartphone magnetic and light sensors. *IEEE International Conference on Communications*, pages 1–6, July 2018.

[70] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv:1506.00019*, October 2015.

[71] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555v1*, December 2014.

[72] Pei Jiang, Yunzhou Zhang, Wenyan Fu, Huiyu Liu, and Xiaolin Su. Indoor mobile localization based on Wi-Fi fingerprint's important Access Point. *International Journal of Distributed Sensor Networks*, 2015:1–8, April 2015.

[73] Dong Yu and Jinyu Li. Recent progresses in deep learning based acoustic model. *IEEE/CAA Journal of Automatica Sinica*, 4(3):396 – 409, July 2017.

[74] Jos Ramiro Martnez de Dios, Anibal Ollero, Francisco Jos Fernndez, and Carolina Regoli. On-line RSSI-range model learning for target localization and tracking. *Journal of Sensor and Actuator Network*, 6:1–19, August 2017.

[75] Shuai Han, Yi Li, Weixiao Meng, Cheng Li, Tianqi Liu, and Yanbo Zhang. Indoor localization with a single Wi-Fi Access Point based on OFDM-MIMO. *IEEE Systems Journal*, 13:964 – 972, March 2019.

[76] Xuyu Wang, Lingjun Gao, and Shiwen Mao. CSI phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet of Things Journal*, 3:1113 – 1123, December 2016.

[77] X. Wang, X. Wang, and S. Mao. Deep convolutional neural networks for indoor localization with CSI images. *IEEE Transactions on Network Science and Engineering*, 7(1):316–327, Jan 2020.

[78] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Sub-nanosecond Time of Flight on commercial Wi-Fi cards. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 121–122, August 2015.

[79] Fabio Ricciato, Savio Sciancalepore, Francesco Gringoli, Nicol Facchi, and Gennaro Boggia. Position and velocity estimation of a non-cooperative source from asynchronous packet arrival time measurements. *IEEE Transactions on Mobile Computing*, 17:2166 – 2179, January 2018.

[80] Xinrong Li and K. Pahlavan. Super-resolution TOA estimation with diversity for indoor geolocation. *IEEE Transactions on Wireless Communications*, 3:224 – 234, January 2004.

[81] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. The Nexmon firmware analysis and modification framework: Empowering researchers to enhance Wi-Fi devices. *Computer Communications ScienceDirect*, 129:269 – 285, September 2018.

[82] Shuyu Shi, Stephan Sigg, Lin Chen, and Yusheng Ji. Accurate location tracking from CSI-based passive device-free probabilistic fingerprinting. *IEEE Transactions on Vehicular Technology*, 67:5217 – 5230, June 2018.

[83] LeCun Y, Bottou L, Bengio Y, and et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, November 1998.

[84] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Proc. 7th Int. Conf. Document Anal. Recognit.*, pages 958 – 963, 2003.

[85] James D. Broesch. *Applications of DSP*. ScienceDirect, 2008.

[86] M. Valente, C. Joly, and A. de La Fortelle. An LSTM network for real-time odometry estimation. *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1434–1440, June 2019.

[87] C. Luo, L. Cheng, M. C. Chan, Y. Gu, J. Li, and Z. Ming. Pallas: Self-bootstrapping fine-grained passive indoor localization using wifi monitors. *IEEE Transactions on Mobile Computing*, 16(2):466–481, 2017.

[88] Y. Duan, K. Lam, V. C. S. Lee, W. Nie, K. Liu, H. Li, and C. J. Xue. Data rate fingerprinting: A wlan-based indoor positioning technique for passive localization. *IEEE Sensors Journal*, 19(15):6517–6529, 2019.

[89] Mikkel Baun Kjrgaard. Indoor location fingerprinting with heterogeneous clients. *IEEE Sensors Letters*, (1):31–43, 2010.

[90] W. Cheng, K. Tan, V. Omwando, J. Zhu, and P. Mohapatra. RSS-Ratio for enhancing performance of RSS-based applications. In *2013 Proceedings IEEE INFOCOM*, pages 3075–3083, 2013.

[91] A. Mahtab Hossain, Y. Jin, W. Soh, and H. N. Van. Ssd: A robust rf location fingerprint addressing mobile devices' heterogeneity. *IEEE Transactions on Mobile Computing*, 12(1):65–77, 2013.

[92] J. Park, D. Curtis, S. Teller, and J. Ledlie. Implications of device diversity for organic localization. In *2011 Proceedings IEEE INFOCOM*, pages 3182–3190, 2011.

[93] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik Rye, and Dane Brown. A study of MAC address randomization in mobile devices and when it fails. *Proceedings on Privacy Enhancing Technologies*, 2017, Mar 2017.

[94] S. S. Sawwashere and S. U. Nimbhorkar. Survey of RTS-CTS attacks in wireless network. In *2014 Fourth International Conference on Communication Systems and Network Technologies*, pages 752–755, 2014.

[95] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy. A CNN-LSTM quantifier for single access point csi indoor localization. *arxiv*, pages 1–11, Apr 2020.

[96] Libtins packet crafting and sniffing library. `http://libtins.github.io/`.

[97] Gather CSI (channel state information) frames with the use of an esp32 wifi chip. `https://github.com/jonathanmuller/ESP32-gather-channel-state-information-CSI-`.