Machine learning for indoor positioning based on received signal strength

Felis Dwiyasa

2018

Felis Dwiyasa. (2018). Machine learning for indoor positioning based on received signal strength. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/89317>

<https://doi.org/10.32657/10220/46235>

MACHINE LEARNING FOR INDOOR POSITIONING BASED ON RECEIVED SIGNAL STRENGTH

FELIS DWIYASA

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2018

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof Lim Meng Hiot, for his guidance and valuable feedback throughout my study. Conducting an independent research is a whole new experience for me, which I found tough and seemed impossible at times, but his support and encouragement motivated me to overcome the challenges I had.

I would also like to thank the Nanyang Technological University, Singapore, for providing me the research scholarship and various forms of financial support, and for the learning opportunities that equipped me with research fundamentals and technical knowledge.

To Mr. Kuan Yeh Cheang from 1RWave LLP and his colleagues, thank you for helping me in my early Ph.D. years with the active RFID hardware and software and for exposing me to the industrial perspective and real applications of active RFID.

To all laboratory staff and friends in IC Design II Lab, Control System Lab, and Communication Research I Lab, thank you for your friendliness and helpfulness.

Last but not least, I would like to thank my family and friends for their love and encouragement throughout this journey. A special thanks is dedicated to my husband for his support through thick and thin.

Table of Contents

Acknowledgements.....	i
Table of Contents	ii
List of Figures.....	vi
List of Tables.....	ix
Abstract.....	xi
Chapter 1 Introduction.....	1
1.1 Problem significance	1
1.2 Objective	7
1.3 Contributions of the thesis.....	8
1.4 Organization of the thesis.....	9
Chapter 2 Literature Review.....	11
2.1 RFID overview.....	11
2.2 Radio propagation behavior	14
2.3 Radio propagation model	16
2.3.1 Free-space path loss	16
2.3.2 Log-distance.....	17
2.3.3 Ray-tracing.....	18

2.4	RF-based positioning metric	20
2.4.1	Power-based metric	21
2.4.2	Time-based metric.....	23
2.4.3	Angle-based metric	25
2.5	Indoor positioning technique.....	26
2.5.1	Geometric approach	26
2.5.2	Location fingerprinting	27
2.5.3	Probabilistic approach.....	33
2.5.4	Handling device heterogeneity	35
2.5.5	Fusion approach	38
Chapter 3	Location Fingerprinting with Continuous Output.....	41
3.1	Introduction	41
3.2	Location fingerprinting	43
3.2.1	Nearest neighbor	45
3.2.2	LANDMARC.....	46
3.2.3	LEMT.....	48
3.3	Methodology of CO-LEMT	52
3.4	Performance evaluation.....	56
3.4.1	Experimental setup.....	56

3.4.2	Performance result under spatial variation	60
3.4.3	Performance result under spatial-temporal variation	66
3.5	Summary	72
Chapter 4	Extreme Learning Machine for Location Fingerprinting.....	73
4.1	Introduction	73
4.2	Extreme Learning Machine	75
4.3	Location fingerprinting translation framework	77
4.4	Methodology of ELM-based LEMT	80
4.5	Performance evaluation.....	82
4.5.1	Experimental setup.....	82
4.5.2	Selection of parameters	83
4.5.3	Performance result under spatial variation	87
4.5.4	Performance result under spatial-temporal variation	92
4.6	Summary	97
Chapter 5	Substitution of Reference Fingerprints	98
5.1	Introduction	98
5.2	Open-access datasets	100
5.3	Dataset description and data adaptation	103
5.4	Performance evaluation.....	107

5.4.1	Accuracy	107
5.4.2	Floor detection	110
5.4.3	Training time and testing time	112
5.5	Summary	114
Chapter 6	Conclusions and Future Work.....	116
6.1	Conclusions	116
6.2	Recommendations for future work.....	118
Author's Publications.....		121
Bibliography		122

List of Figures

Figure 2.1: Line-of-Sight (LOS) multipath propagation.....	15
Figure 2.2: Non-Line-of-Sight (NLOS) multipath propagation	16
Figure 2.3: Two-ray model's propagation paths	19
Figure 2.4: Free-space path loss and two-ray path loss over distance	20
Figure 3.1: Continuous and discrete output space in location fingerprinting ...	42
Figure 3.2: Sparse training points in discrete-output nearest neighbor	46
Figure 3.3: An example of M5 model tree representing functional relationship between RSS of a tracking device, $sj(i)$, and RSS of reference devices, $rkj(i)$	50
Figure 3.4: Block diagram of CO-LEMT	53
Figure 3.5: Site layout for performance evaluation of location fingerprinting.....	59
Figure 3.6: Cumulative error distribution of location fingerprinting for Dataset A under spatial variation.....	62
Figure 3.7: Cumulative error distribution of location fingerprinting for Dataset B under spatial variation	63
Figure 3.8: Cumulative error distribution of location fingerprinting for Dataset C under spatial variation	64
Figure 3.9: Cumulative error distribution of location fingerprinting for Dataset A under spatial-temporal variation	68

Figure 3.10: Cumulative error distribution of location fingerprinting for Dataset B under spatial-temporal variation	69
Figure 3.11: Cumulative error distribution of location fingerprinting for Dataset C under spatial-temporal variation	70
Figure 4.1: Extreme Learning Machine (ELM) architecture	75
Figure 4.2: Training time of CO-LEELM under different numbers of hidden neurons, N_h , and different regularization parameter, C	84
Figure 4.3: Testing time of CO-LEELM under different numbers of hidden neurons, N_h , and different regularization parameter, C	84
Figure 4.4: Accuracy of CO-LEELM for Dataset A under different numbers of hidden neurons, N_h , and different regularization parameter, C .	85
Figure 4.5: Accuracy of CO-LEELM for Dataset B under different numbers of hidden neurons, N_h , and different regularization parameter, C .	85
Figure 4.6: Accuracy of CO-LEELM for Dataset C under different numbers of hidden neurons, N_h , and different regularization parameter, C .	86
Figure 4.7: Cumulative error distribution of ELM-based methods for Dataset A under spatial variation.....	88
Figure 4.8: Cumulative error distribution of ELM-based methods for Dataset B under spatial variation	89
Figure 4.9: Cumulative error distribution of ELM-based methods for Dataset C under spatial variation	90
Figure 4.10: Cumulative error distribution of ELM-based methods for Dataset A under spatial-temporal variation.....	93

Figure 4.11: Cumulative error distribution of ELM-based methods for	
Dataset B under spatial-temporal variation	94
Figure 4.12: Cumulative error distribution of ELM-based methods for	
Dataset C under spatial-temporal variation	95
Figure 5.1: Site layout of long-term Wi-Fi dataset	104
Figure 5.2: Configuration of training and testing data/reference for long-	
term Wi-Fi dataset	105
Figure 5.3: Accuracy of long-term Wi-Fi dataset	109
Figure 5.4: Cumulative error distribution of long-term Wi-Fi dataset.....	110
Figure 5.5: Floor detection success rate of long-term Wi-Fi dataset	111

List of Tables

Table 3.1: Types of information used in location fingerprinting system	44
Table 3.2: Comparison of location fingerprint input, output, and weakness	52
Table 3.3: Datasets and test scenarios for location fingerprinting	58
Table 3.4: Accuracy of location fingerprinting under spatial variation	61
Table 3.5: Training time and testing time of location fingerprinting under spatial variation	65
Table 3.6: Accuracy of location fingerprinting under spatial-temporal variation	67
Table 3.7: Training time and testing time of location fingerprinting under spatial-temporal variation	71
Table 4.1: Configuration of ELM-based location fingerprinting methods	78
Table 4.2: Description of proposed ELM-based location fingerprinting methods	82
Table 4.3: Final choice of ELM parameters.....	86
Table 4.4: Accuracy of ELM-based methods under spatial variation	87
Table 4.5: Processing time of ELM-based methods under spatial variation.....	91
Table 4.6: Accuracy of ELM-based methods under spatial-temporal variation	92
Table 4.7: Training time and testing time of ELM-based methods under spatial-temporal variation	96

Table 5.1: Open-access datasets for RSS-based indoor positioning	102
Table 5.2: Accuracy of long-term Wi-Fi dataset	108
Table 5.3: Training time and testing time of long-term Wi-Fi dataset	113

Abstract

Indoor positioning is a key technology enabler for various smart systems that require location-based optimization and automation. Despite its potential, the field of indoor positioning faces numerous challenges in terms of accuracy, precision, computational complexity, power consumption, robustness, scalability, and cost.

This thesis proposes machine learning approach to improve indoor positioning system that makes use of Received Signal Strength (RSS) positioning metric, in terms of accuracy, processing time, and robustness. We consider machine learning approach to be a potential alternative solution due to its ability to model nonlinear functions without *a priori* assumptions on the distribution of the data and the noise.

We present several improvements in location fingerprinting approach with the objective of reducing the number of training data, improving robustness, and decreasing the length of processing time. We combine two existing location fingerprinting methods, which are Location Estimation using Model Tree (LEMT) and LANDMARC, to improve the accuracy and robustness of the location fingerprinting when training points are sparsely distributed. Performance evaluation shows that the proposed continuous-output modification of LEMT achieves 0.5 m to 2 m reduction in the 90th and 95th error percentile.

LEMT method and its derivatives, however, have long training time and testing time because these methods rely on M5 model tree which is computationally intensive. The M5 model tree is used in LEMT to model the functional relationship between reference devices and tracking device. To address this problem, we propose the use of Extreme Learning Machine (ELM) regression model in place of M5 model tree to shorten the processing time. We found that incomplete data in the RSS fingerprint can either be imputed with a low RSS or filtered out, without negatively affecting the accuracy of ELM regression. With the use of ELM, the location fingerprinting method experiences 15 to 40 times improvement in training time and 6 to 10 times improvement in testing time, with accuracy degradation around 0.1 m to 0.5 m. Depending on the application involved, this may be considered reasonable or acceptable.

We also evaluated the performance of the proposed methods using an open-access benchmark dataset containing no reference device, measured across 15 months. When reference devices are not present in the environment, we can use a subset of training fingerprints as reference data in place of the fingerprints of reference devices. Since we only use a subset of fingerprints, we reduced the amount of work in collecting repeated training data. Our proposed method produces superior accuracy across all months, indicating its ability to adapt to temporal variation despite having minimal amount of repeated training data, especially for the 11th month and beyond when the benchmark methods start to experience significant accuracy degradation.

Chapter 1

Introduction

1.1 Problem significance

Recent advances in positioning technology have resulted in significant lifestyle transformation. Since the introduction of Global Positioning System (GPS) satellites for non-military use in the late 1990s, positioning system has become an integral part of human life in most parts of the world. Combined with global maps and global time, GPS helps people to know where they are, how to go to a certain place, when a vehicle or someone arrives at a place, and many more. GPS allows people to find buildings and places of interests, saving them the time and effort to move from one place to another. By equipping cars or buses with GPS receivers, transportation service providers can track the real-time positions of their vehicles, provide an accurate arrival timing prediction, and perform various data analyses to improve the efficiency of their service.

GPS works best in outdoor environment where there is no obstacle in the signal propagation paths between the GPS receiver and the satellites. Thus, GPS has been considered as the best technological solution for various outdoor positioning applications such as transportation, logistics, and navigation systems. However, GPS is not suitable for indoor use as its accuracy degrades significantly in indoor

environment because the paths between satellites and GPS receiver are typically obstructed by the structure of buildings and various indoor objects.

Indoor positioning technology could potentially change human behaviors even more than outdoor positioning. It is an enabler for many potential real-life applications, such as indoor navigation within buildings, personal assistant for people remembering things or events, smart library or smart workplace where any object can be easily located, and many more. In logistics, indoor positioning allows real-time tracking and automation of the movements of the goods, reducing the effect of human error and negligence. The tracking data may also be analyzed for further resource or process optimization.

Unlike outdoor positioning methods which tend to be quite accurate and reliable with GPS, the best technology solution for indoor positioning system is still nowhere near from being conclusive. Various technologies, including Radio Frequency (RF), ultrasound, infrared vision, motion sensor, and fusion approach, have been explored in the hope of obtaining a reliable solution for indoor positioning. Each of these technologies has its own advantages and limitations.

Portable devices such as mobile phones and tablets are typically embedded with RF technology such as Wi-Fi and Bluetooth, which are mainly used for data transfer purpose. Newer wireless technologies such as Radio Frequency Identification (RFID), Ultra-Wideband (UWB), and Near Field Communication (NFC) have also emerged in the market to serve various unique purposes such as

item tagging, security access, banking transactions, and many more. In addition, indoor positioning can also be performed using these wireless technologies.

There are several aspects to consider in designing and choosing technology and algorithm for indoor positioning systems. As surveyed by Liu et al., those aspects include accuracy, precision, computational complexity, power consumption, robustness, scalability, and cost [1]. To date, there is no single technology which delivers the best performance in all aspects. For example, Wi-Fi and Bluetooth technologies are readily available in personal devices, making it cost-effective for large-scale implementation. However, they are quite power hungry with moderate accuracy, ranging from few meters to tens of meters depending on the environment. Similar technology like RFID has much lower power consumption, but its accuracy is about the same with Wi-Fi and Bluetooth. Higher accuracy is possible using more complex algorithms, but high computational complexity directly translates to longer processing time and higher power consumption. Fusion approach using wireless technology and motion sensor has better accuracy, but its power consumption increases significantly due to the need of sampling data continuously from the motion sensor. On the other hand, UWB technology is known to have accuracy of up to 10 cm and low power consumption, but the cost is about tenfold higher, making it less affordable for large scale deployment. These problems present as challenges as well as opportunities for further research.

The radio signal of wireless transceivers can produce positioning metrics that are correlated in some way with physical space. When such correlation is known, we

can estimate the physical location of the wireless transceiver. The process of approximating the correlation, however, is often not straightforward because the observed positioning metrics are often noisy, non-linear, and time-varying. This phenomenon is caused by the radio propagation behavior in indoor environment that is complex and dynamic in nature.

The observed data in indoor positioning system typically experiences 3 types of signal variations:

1. Spatial variation refers to the different propagation effects experienced by radio signal at different locations. Spatial variation may present itself in the form of large-scale and small-scale spatial variations. Large-scale spatial variation refers to different propagation effects at places separated by a large distance, whereas small-scale spatial variation refers to a phenomenon where small moves in the order of wavelength can cause extreme signal fluctuation [2]. Moreover, radio propagation is affected by the types of material on which the device is attached [3], and, for wearable devices, by the presence and orientation of the user's body [4] [5].
2. Temporal variation refers to the time-variant characteristic of the radio propagation effect. This is likely due to abrupt changes, such as the adjustment of furniture placements or building layout, as well as small regular changes such as door's opening and closing, human activities within building, and temporary signal interference from other wireless devices [6]. It is important to consider temporal variation in indoor

positioning system because indoor environments are likely to change over time.

3. Device variation refers to the condition where one device behaves differently from another device. Different brands of wireless devices of the same technology may have different product specifications [7], particularly on the antenna radiation pattern, transmitter gain, and receiver sensitivity. They may also have unique radiation patterns when connected to different antenna or enclosed with different packaging [8]. Device variation can cause a positioning method to be unreliable if the method requires the devices being used to be identical.

Physical locations can be estimated from positioning metrics using either parametric or non-parametric machine learning. In parametric model, it is assumed that the distribution of the data or noise is known. For example, noise is often assumed to follow zero-mean Gaussian probability distribution with a known variance. Another common assumption that is often used is that the mapping function between the observed data and the position estimate follow a certain function, such as linear or logarithmic. The problem of such assumptions is that, in reality, the exact characteristics of those parameters are not known, and the assumption may not hold true. In such cases, parametric models that work perfectly on simulated data would behave strangely on real data. Non-parametric machine learning, on the other hand, relies on little or no *a priori* assumptions of the characteristics of the parameters. Instead of making such assumptions, it

infers the characteristics of the data from the data itself. The “non-parametric” term in non-parametric machine learning does not mean that the machine learning does not contain any parameters. Instead, this model typically contains a large number of parameters that are adjusted automatically when training data are fed into the model. During training process, the model puts more weights on important parameters and less weights on unimportant parameters so that the training output could fit the training input. Non-parametric machine learning has been known to be able to deal with noisy and nonlinear data better than parametric machine learning.

Despite its benefits, non-parametric machine learning model is generally slower than parametric model because it has a larger number of parameters to adjust. If the training speed is too slow, it would be difficult to scale the model to handle a bigger dataset. Also, since non-parametric model requires a training process where the parameters of the model are adjusted according to training data, the required amount of training data is also a crucial aspect to consider. It is necessary to have training data that could represent the overall problem with a limited number of collected samples. If the number of samples is too small, the model may not represent the problem correctly. In contrast, as the number of samples becomes larger, the collection of training samples would be more tedious and more prone to human error.

Estimating location coordinates from positioning metrics may involve more than one model, and each of those models can either be parametric or non-parametric.

This can be observed in two major approaches in indoor positioning – geometric approach and location fingerprinting approach. In geometric approach, there are at least 2 separate models, a radio propagation model that translates positioning metrics into distance vector, and a model that translates the distance vector into location coordinate. The radio propagation model can either be parametric or non-parametric, depending on whether the functional mapping and noise distribution of the positioning metrics are set based on assumption or concluded from training data. The second approach, which is location fingerprinting, contains a model that record or learn the location fingerprints, which is typically non-parametric, and a model that concludes the location coordinate.

1.2 Objective

The objective of this research is to develop indoor positioning models that can learn the environmental condition from a limited amount of training data, with better accuracy and shorter training time than the existing indoor positioning methods.

Considering that indoor positioning is a complex problem that might not be adequately represented by simulation data, throughout this research we use real measurement data collected from active RFID devices to evaluate the performance of our proposed methods. Also, to make sure that the methods work

in another setting, we conducted performance evaluation using an open-access data of Wi-Fi RSS measurements.

1.3 Contributions of the thesis

The main contributions of this thesis are as follows:

- We combined the methodology of two existing location fingerprinting methods, which are LANDMARC and Location Estimation using Model Tree (LEMT), to form a new continuous-output location fingerprinting method. This method is found to have better accuracy and robustness under various conditions.
- We constructed a systematic framework to map location fingerprinting methods into algorithms that use Extreme Learning Machine (ELM) which could potentially achieve lower computational complexity and shorter training time that is suitable for real-time implementation.
- We proposed the use of ELM as a functional relationship model in place of M5 model tree in LEMT. Incomplete data from reference devices can be imputed with a low RSS value without degrading the accuracy of the ELM-based method. Our proposed method is found to have shorter processing time and better accuracy than the existing LEMT technique.

- We proposed the use of a subset of training fingerprints in place of reference devices, allowing our proposed methods to be applied in an environment setup where reference devices are not available. Although the absence of reference devices requires us to repeat the training data collection to be able to capture temporal variation, our proposed method produces better accuracy than NN and LEMT with smaller number of repeated training data.

1.4 Organization of the thesis

Chapter 2 contains a brief overview of RFID and the basics of indoor positioning methods, specifically those that are based on RF technology.

Chapter 3 presents a continuous-output location fingerprinting method that combines Location Estimation using Model Tree (LEMT) and LANDMARC. With continuous output, training data can be collected in sparsely distributed locations, allowing the number of training data to be reduced. The proposed method also uses reference devices to adapt to temporal variation. Performance of the proposed method and the existing location fingerprinting methods are evaluated using real indoor data collected under various conditions.

Chapter 4 presents a new framework for translating location fingerprinting methods into ELM-based methods. An improved version of continuous-output

location fingerprinting from Chapter 3 is also presented and evaluated using real indoor data.

Chapter 5 presents a performance evaluation of the proposed location fingerprinting methods presented in Chapter 3 and Chapter 4 using an open-access data of Wi-Fi RSS fingerprints. Since the system setup in the open-access data does not use any reference devices, we replace the reference fingerprints with a subset of training fingerprints.

Finally, we present our conclusions and recommendations for future work in Chapter 6.

Chapter 2

Literature Review

2.1 RFID overview

Radio Frequency Identification (RFID) is a wireless technology which has been gaining popularity in recent years. RFID has been widely used in various applications that require identification, positioning, and tracking functionality. It has been used in many commercial applications such as warehouse management system, inventory tracking, people tracking, building management, and parking control.

A typical RFID configuration consists of one or several readers and many tags. Readers are equipped with more functionalities than tags and are less power-constrained as they are typically plugged into power outlets, whereas tags are usually more portable and power-constrained than readers. RFID readers are often equipped with a capability to build a predefined or auto-forming network.

RFID has relatively low power consumption. Most RFID batteries can last for several months or even years, depending on the operational duty cycle. Some RFID tags can even operate without battery. This low power characteristic makes RFID a potential technology solution for applications that cater to user mobility and minimum maintenance effort.

Based on how the tags are powered up and energized, RFID can be categorized into passive, semi-active, and active RFID [9] as follows:

1. Passive RFID

Passive tags have no power source attached. The power comes from a nearby reader that provides electrical power through inductive coupling for powering up the circuits and sending signals. The reader, which is also called interrogator, initiates the tag's interrogation besides providing power. Most passive tags contain uniquely shaped conductive wires which reflect electromagnetic patterns field generated by a reader. Passive tags contain a minimum amount of electronic circuits, making it the cheapest and most robust type of RFID. However, passive tags have a very limited access range because of the inductive coupling requirement.

2. Semi-active RFID

Semi-active tags have internal power sources, usually in the form of batteries, to power up the circuits. However, tags do not initiate a communication until a reader tries to reach them. Right after a reader initiates communication link, tags can use the reader's power to respond. The communication range of semi-active tags is usually slightly longer than passive tags.

3. Active RFID

Active tags use the internal power sources to power up their circuits as well as to send signal to readers. For this type of RFID, tags usually send ping signal periodically while readers continuously listen for signal from any tags within its range. Depending on the system configuration and power consideration, readers may or may not send acknowledgment signal to respond to the tag. The communication range of active tags is larger than passive and semi-active tags. One of the most significant advantages of active tags is its large access range, which can reach tens to hundreds of meters. For positioning purpose, it helps to reduce the density of readers that need to be installed and therefore it could reduce system costs. Active tags can also share the same power source with other sensors and be programmed to form wireless sensor networks.

RFID systems can be configured in several ways, depending on the type of topology of the positioning system that we choose. A reader or a tag can be functionally configured as a signal transmitter, a signal receiver, a distance measuring unit, or a unit performing some of those tasks. One of the most commonly used configurations is to use tags as objects to be identified, positioned, and tracked by readers [10] [11]. Since tags are usually much cheaper than readers, it is more economical to scale up the number of tracked objects when the objects are equipped with tags instead of readers. Systems that switch those roles, where the readers are attached to the tracked objects, is usually aimed

at tracking expensive equipment. An example for such configuration is Smart Floor [12], where RFID tags are placed on the floor and used to track a robot equipped with an RFID reader.

2.2 Radio propagation behavior

Radio signal has been known to have complex propagation behaviors such as absorption, reflection, diffraction, and scattering [13]. When propagating from one place to another, radio signal tends to dampen over distance. This is known as attenuation effect. When modeling attenuation effect, it should be noted that the amount of attenuation is highly influenced by the type and the thickness of the propagation medium.

In real environment, radio signal typically need to penetrate several mediums, such as air, concrete, and wood that simultaneously exist within the propagation path. The characteristics of the medium may also be influenced by the atmospheric condition.

Besides attenuation effect, objects around the radio transmitter and receiver may also cause radio signal to be reflected, diffracted, and scattered. As a result, the signal would experience nonlinear distortions when observed at the receiver side.

Radio propagation typically involves more than one propagation path. When a transmitted signal reaches the receiver through a direct path and several indirect

paths, the receiver observes them as a superposition of multiple signals. This phenomenon is known as multipath effect.

Direct path is a propagation path where radio signal from a transmitter directly reaches its receiver without being distorted by any object. In contrast, radio signal traveling in indirect path is either reflected, diffracted, or scattered by surrounding objects before arriving at the receiver. Indirect path requires a longer propagation time than direct path and appears as an echo of the direct path signal with additional changes in signal power and phase.

In Line-of-Sight (LOS) condition illustrated in Figure 2.1, the received signal contains the signal coming from a direct path in addition to some indirect paths. This happens when there is no significant obstruction in between the transmitter and receiver. In Non-Line-of-Sight (NLOS) condition as shown in Figure 2.2, the received signal only contains indirect paths and no direct path.

Figure 2.1: Line-of-Sight (LOS) multipath propagation

Figure 2.2: Non-Line-of-Sight (NLOS) multipath propagation

2.3 Radio propagation model

2.3.1 Free-space path loss

The simplest model of radio propagation is free-space path loss (FSPL) model [13]. This model assumes that the radio signal propagates through free-space, there is a line of sight between the transmitter and receiver, and there is no disturbance caused by any nearby objects.

The formulation of FSPL is given by:

$$PL = \left(\frac{4\pi d}{\lambda} \right)^2 = \left(\frac{4\pi d f}{c} \right)^2 \quad (2.1)$$

where d is the distance of separation in meter, c is the speed of light which is approximately 3×10^8 m/s, and λ is the wavelength in meter which is inversely proportional to the signal frequency, f , in Hertz.

The free-space path loss gives a good insight on the relationship between distance, frequency, and signal power. However, there are several important considerations for real environmental condition. Firstly, it assumes that the transmission occurs in free-space. It does not consider air, water, or any other media that would exist on earth. Secondly, it assumes that there is only one direct path, possible only if there is nothing around the transmitter and receiver. In reality, various objects may exist around the propagation path and affect the propagation of radio signal. Therefore, free-space propagation model is usually inadequate to model radio propagation in real environment.

2.3.2 Log-distance

Log-distance model takes the propagation medium into account [13]. The main difference between this model and free-space path loss model lies in the exponential factor given in the following equation:

$$PL = \left(\frac{4\pi d}{\lambda} \right)^N = \left(\frac{4\pi df}{c} \right)^N \quad (2.2)$$

where N is a real positive value representing exponential factor that depends on the propagation medium. The exponential factor can be estimated through empirical data [14] [15] or determined beforehand by assuming the type of propagation environment.

Indoor propagation model proposed by International Telecommunication Union (ITU) [16] uses a similar approach as the log-distance model. The exponential factor in the path loss equation depends on the environment factor. In addition, obstructions such as walls and floors are also to be considered into the path loss calculation as given by

$$L_{total}(dB) = 20\log_{10}f + N\log_{10}d + L_f(n) - 28 \quad (2.3)$$

where N is path loss coefficient, $L_f(n)$ is the floor penetration loss in decibel which depends on n the number of floors, frequency f , and the type of building. The constant offset of 28 dB in the ITU indoor propagation model is obtained through empirical study in various indoor environment. Likewise, other parameters in this model, which include path loss coefficient and floor penetration loss, are fixed constants formulated through empirical study.

2.3.3 Ray-tracing

Free-space path loss and log-distance model provide a rough estimate on signal attenuation and therefore can be used to estimate the coverage area for network planning purpose. However, problem arises when these models are being used for positioning system.

Due to radio propagation behavior, signal arriving at the receiving end does not purely consist a direct path between the transmitter and the receiver. With positive

and negative interference caused by the indirect paths, signal strength would not monotonically decrease over distance as depicted by free-space path loss model and log-distance model.

Ray tracing model [17] [18] considers signal reflection paths in addition to the direct path. The simplest ray tracing model is the two-ray model that can be used to emulate direct path signal and interference signal from ground-reflection path as illustrated in Figure 2.3. The number of rays can be larger than 2 for more complex environment. The higher the number of rays to consider, the more complex it is to compute.

Figure 2.3: Two-ray model's propagation paths

Propagation path loss for a two-ray model shown in Figure 2.4 suggests that RSS is multi-modal and does not monotonically decrease with distance. If this holds true, we cannot infer a specific distance simply by mapping the RSS into distance as in free-space or log-distance path loss model.

It has been verified in a work involving probabilistic approach that ray-tracing model is found to be closer to reality than conventional log-distance model [19].

Figure 2.4: Free-space path loss and two-ray path loss over distance

2.4 RF-based positioning metric

There are 3 common types of positioning metric for indoor positioning that use RF signal as its measurement technique: 1) power-based, 2) time-based, and 3) angle-based. Each metric has its unique characteristics and is obtained through different means.

2.4.1 Power-based metric

2.4.1.1 Signal strength

Signal strength, which is also known as Received Signal Strength (RSS) or Received Signal Strength Indicator (RSSI), is a power-based metric that is measured at the receiver side in the form of signal power. Measurement of RSS requires no additional hardware and is typically an in-built function in wireless devices as it plays a vital role during the establishment of data communication link. RSS-based positioning can be applied on different technologies, such as RFID, Wi-Fi, Bluetooth, and UWB.

As discussed in the previous section, radio signal tends to be weaker when the distance between transmitter and receiver increases. This tendency is often modeled as a regression problem [20], where RSS is mapped into distance by using various radio propagation models described in Section 2.3.

RSS metric however, suffers from multipath propagation which changes the amplitude of the signal in a complex way. When multipath effect occurs, radio signal arriving at the receiver side would be noisy and have a nonlinear relationship with its physical distance. Some ways to mitigate the measurement noise includes applying moving average filtering on multiple samples [21], truncated Singular Value Decomposition (SVD) technique [22], scrambling method [23], logarithmic spectrum averaging [24], or Karhunen-Loeve (KF) transformation [25].

2.4.1.2 Channel response

Another type of power-based metric that can be used for positioning is channel response. This metric can be obtained from wireless system that uses wideband signal such as Wi-Fi system that is based on Orthogonal Frequency Division Multiplexing (OFDM) modulation. On the receiver side, the received signal is compared with the transmitted preamble or pilot subcarriers, producing a channel response estimation that measures the amplitude and phase change of smaller chunks of signal bandwidth. Another way to obtain this metric is to transmit a very short impulse signal, as used in Ultra-Wideband (UWB) system, and analyze the impulse response on the receiver side. The observed metric is known as Channel Impulse Response (CIR) in time domain [26] or Channel State Information (CSI) in frequency domain [27].

Channel response estimation provides a finer-grain information that captures the combined effect of multipath propagation better than RSS information that represents the signal power across the bandwidth as a whole [26] [28]. It is suggested that the accuracy of channel-response based positioning can be even higher if we use larger bandwidth [29], multiple antennas [30], or deep learning algorithm [27] [30].

Despite its potential, the use of channel response as a positioning metric for communication systems in current literatures are limited to systems that are based

on OFDM and UWB signal. Its use in other types of modulation systems such as in Direct Sequence Spread Spectrum (DSSS), which is commonly used in active RFID, is not explored yet. Channel response is also not possible to obtain in narrowband signal. Thus, wireless technologies that use channel response as its positioning metric must have center frequency that is high enough to accommodate large bandwidth.

2.4.2 Time-based metric

Because radio signal travels at the speed of light, the traveling time required by radio signal to reach one point from another point can be used to estimate how far the two points are separated. Time of Arrival (ToA) is a time-based metric that measures the time required for the radio signal to travel from transmitter to receiver [31]. To achieve this, the transmitter and receiver need to have precise synchronized clocks. Because high precision oscillators are expensive and continuous synchronization is often impractical, this method is not preferred for low-cost applications.

Roundtrip ToA, which is also known as Roundtrip Time of Flight (RToF), avoids the need of clock synchronization by having only one device actively sending signal and receiving signal reflected back by the responder. With this method, clock synchronization is not required and the high-precision clock is required only for the reader. This metric is often used in passive RFID, where a reader

transmits signal to a transponder that backscatters the signal to the reader [32]. However, in addition to the time required by the signal for a roundtrip travel, there is also some processing time required by the responder which can only be neglected if it is small enough compared to the traveling time. The application of this method is quite limited because passive RFID has very limited access range, which is only up to few centimeters. Moreover, it relies on one reader to communicate with the tag and to measure the distance at a time.

Another possible time-based metric is Time Difference of Arrival (TDoA) which measures the time difference of the signal to travel from a transmitter to several synchronized receivers [33]. With this solution, multiple receivers can work at the same time to produce a position estimate. Precise oscillators only need to be installed in stationary RFID readers or WLAN access points, while lower-cost oscillators can be used in the transponder or mobile devices without potentially causing measurement error.

Time-based metric has some limitations and drawbacks. Li and Pahlavan studied that acceptable time resolution can only be achieved using wideband signal, of which a finer-grain measurement can be performed by observing the channel response [34]. Like power-based metric, time-based metric also suffers from multipath problem. In multipath propagation, a radio signal travels through several different paths, causing it to be received as multiple signals arriving at slightly different time delay. When the signal coming in a direct path is not the strongest signal received, the estimated traveling time would be detected

incorrectly. Identifying Non-Line-of-Sight (NLOS) conditions using neural network as channel responses that have a high delay spread and compensating the NLOS distance estimates by subtracting them with the statistics of distance estimation error. This was found to produce a better ToA positioning accuracy [35]. Since ToA benefits from larger bandwidth, UWB technology which occupies signal bandwidth more than 500 MHz would be a better choice for implementing ToA [36]. Some UWB indoor positioning devices such as Zebra's DART¹ and Decawave's Scensor² seem to exploit this characteristic to achieve a highly accurate positioning.

2.4.3 Angle-based metric

Angle of Arrival (AoA) measures the direction of the signal arrival at the receiver's antenna. The measurement of AoA needs additional hardware in the form of directional antenna [37] [38] or antenna arrays [39] [40]. Due to its antenna requirement, AoA-enabled devices are relatively large and consume more power.

¹DART Ultra-Wideband (UWB) Technology,
<https://www.zebra.com/us/en/solutions/location-solutions/enabling-technologies/dart-uwband.html>

²ScenSor DW1000, <http://www.decawave.com/products/dw1000>

2.5 Indoor positioning technique

2.5.1 Geometric approach

Geometric-based approach can be categorized into proximity, lateration, and angulation methods. Proximity method estimates the location of a tracking device by selecting one anchor device that is most likely to be the nearest, lateration method concludes the location estimate from the distance estimation from multiple anchors, whereas angulation method makes use of the angular information from multiple anchors.

The basic idea of proximity-based localization is to find an anchor device that senses the strongest signal and to assume that the tracking device is located near that anchor device. An example of this proximity-based localization is Smart Floor [12]. The accuracy of this method is highly affected by the density of the anchors. Therefore, a sufficient number of receivers must be deployed to maintain reasonable accuracy. As this method requires many anchors, the cheaper the cost of the anchors, the more feasible it is to implement.

Lateration incorporates distance information which is translated from the received signal power using radio propagation model. The accuracy of lateration method depends greatly on the accuracy of the distance estimates. Since RSS is very sensitive to obstructions, multipath effects, and even antenna orientation, distance estimate may appear shorter or longer than it actually is.

When Least Square (LS) optimization is used in trilateration [41], the objective function of the optimization is to minimize the residue squares of distance error. When the distance error contains outliers, the accuracy of the trilateration will be significantly affected by the inaccurate distance estimates. Least Medium Square (LMS) presented by Li et al. [42], which uses the median value of the residue squares as the objective function, is more robust to outliers than LS, but provides worse of estimation than LS when there is no outlier.

Another known way to tackle outlier problem is through bilateration method [43] [44], where pairs of location candidates produced by bilateration units are selected in a way that the effect of inaccurate distance estimates can be reduced. This method, however, tend to be slower than trilateration as it involves pair-by-pair comparisons of location candidates. Ho and Lee discussed an effort in reducing the processing time of bilateration [45], where the computation of location estimates is performed by applying Heron's formula that is faster than conventional bilateration. This effort, however, focuses on the computation of the location estimates and not on the identification of outliers in location estimates.

2.5.2 Location fingerprinting

2.5.2.1 Nearest neighbor

Early fingerprinting approach infers the user location using Nearest Neighbor (NN) technique, which assumes that locations that are physically close tend to be

close in signal space [4]. This technique involves an offline site survey to record RSS fingerprints at various possible locations, along with the ground-truth location coordinates at which the fingerprints are collected. NN assumes that the most possible location of the tracking device during online testing stage is the location that has the closest matching fingerprint, which is the one that produces the smallest Euclidean distance between the offline and online fingerprint.

NN technique is quite reliable under a static environment. However, when there is a significant change in the environmental conditions, the recorded fingerprints may not represent the radio map of the environment anymore. Offline fingerprints must be recollected to achieve good positioning accuracy. Unfortunately, collecting offline fingerprints repeatedly is tedious and costly, especially if the environment changes quickly and the test-site covers a very large area.

Another variation of NN technique is K-Nearest Neighbor (KNN) where the coordinates of K nearest neighbors are averaged [4], weighted [46], or interpolated using Kriging method [47] to estimate the coordinate of the tracking device.

Kaemarungsi proposed Mahalanobis distance to measure the distance between the online fingerprint and the distribution of offline fingerprints by taking into account the covariance matrix of the distribution of offline fingerprints [48]. Mahalanobis distance reduces to Euclidean distance if the covariance matrix is an identity matrix, or in other words, Mahalanobis distance is a generalized form

of Euclidean distance. It produces a more accurate positioning than Euclidean distance with a trade-off in higher computational complexity.

2.5.2.2 LANDMARC

LANDMARC addresses the problem of temporal variation by using reference points to capture the real-time characteristic of radio map [10]. The position of the tracking device is estimated by finding the nearest neighbors in the signal space of reference tags, compute the weights of each neighbor, and combine the weighted coordinate of the reference devices as the coordinate of the tracking device.

There are some known limitations of the LANDMARC method. By calculating Euclidean distances between fingerprints of different tags, LANDMARC assumes that all tags have a nearly identical gain characteristic. If reference tags or tracking tag have different antenna radiation or gain, which are very common in mass-produced devices, the algorithm would incorrectly detect the matching fingerprint. Another limitation of this method is that it requires reference tags to be placed in a way that the nearest reference tags should not be blocked by walls or other objects in an unbalanced manner. Consequently, a dense placement of reference devices is required, which in turn increases the deployment and maintenance cost.

In Virtual Reference Elimination (VIRE) [49], the accuracy of LANDMARC is improved by adding virtual reference points. RSS at each virtual reference point is obtained by applying linear interpolation to the RSS values of its two adjacent reference devices. Experimental work has shown that when the real reference devices are separated by 1-meter distance, VIRE could achieve 17 to 73 percent accuracy improvement as compared to LANDMARC. The results, however, have not been confirmed for cases where the separation distance is larger than 1 m. With larger distance between reference devices, linear relationship between adjacent reference devices may not apply.

2.5.2.3 LEMT

Yin presented Location Estimation using Model Trees (LEMT) [6], which is a location fingerprinting method that shares the same principle with LANDMARC in the way that it uses reference devices placed at static positions to learn the temporal variation in the environment. The functional relationship between the reference devices and the tracking device is learned from the RSS data collected during the training phase.

LEMT is found to have better accuracy and to require less number of reference points than LANDMARC. It is also less sensitive to the non-uniformity of device hardware and propagation path as compared to LANDMARC. Despite its promising accuracy and robustness, LEMT has a high computational complexity

due to the nature of the M5 model tree used for the functional relationship model. Also, the location estimates produced by LEMT are discrete and limited by the locations of training points.

2.5.2.4 Neural networks

Brunato and Battiti proposed one of the early research efforts that involve neural networks in indoor positioning [50]. Their work compares probabilistic Bayesian approach and several neural network architectures in terms of accuracy and processing time. Despite having much shorter processing time, neural networks are found to produce better accuracy on real indoor data.

Wu, Fu, and Lian configured Support Vector Machine (SVM) as regression module to infer the location coordinates from RSS fingerprints [51]. A mechanism of filtering out sudden changes in location estimation caused by RSS fluctuation is also introduced.

In a large-scale test where the testing area is generally larger than the access range of the transceivers, some RSS fingerprints may contain incomplete data. To address null values that exist in the incomplete data, Ahmad et al. extended the scalability of indoor positioning using Multi-Layer Perceptron (MLP) by applying clustering method [52].

Moreno-Cano et al. proposed the use of Radial Basis Function (RBF) network and particle filters to estimate and track the location of a RFID tag with the assistance of RFID reference tags and IR sensor [53]. This work shows that a combination of several positioning methods and sensors can produce a better accuracy. However, processing time is not part of the performance evaluation of this work.

Although conventional neural networks such as SVM, MLP, and RBF have been able to produce a relatively good accuracy, the training process of those methods relies on iterative backpropagation learning framework. This contributes to a longer training time, which is not favorable in real-time implementation.

The use of Extreme Learning Machine (ELM) [54] for indoor positioning has been introduced in several works. ELM can be configured as a regression or classification module that maps the RSS fingerprints into either location coordinates or location index. Xiao et al. proposed a hybrid approach using NN clustering and ELM, where the clustering is found to produce better accuracy with less processing time than the non-clustered method [55]. Liu et al. introduced Semi-supervised ELM (SELM) to involve unlabeled training data in the training process, which in turn will reduce manual labeling effort during training data collection [56]. To achieve shorter training time, Zou et al. introduced incremental training through Online Sequential Extreme Learning Machine (OS-ELM) [57]. The abovementioned ELM-based methods, however, relies on the assumption that the training data are drawn from the same

distribution, which is not suitable for dynamic environment where temporal variation exists.

2.5.3 Probabilistic approach

Probabilistic model stores the distribution of the positioning metric during the training phase and compares them with the positioning metric obtained during testing phase using a probabilistic technique. Probabilistic model and deterministic model are found to achieve similar positioning accuracy for static objects, but probabilistic model is more accurate than deterministic model for fast-moving objects [58]. In Bayesian inference framework, the *a priori* knowledge of the user's location probability is assumed to be known [59] or estimated from empirical measurement along a predetermined path [60]. Bayesian inference framework can be implemented using Kalman filter [61] or particle filter [62] [63]. Motion model can be included into the framework as Voronoi diagram [62] [64], constant velocity model [63], Monte Carlo localization [65], or Gaussian kernel [66]. Unlabeled information from user traces can be used to refine the radio map using online co-localization [67] or using Hidden Markov Model (HMM) and Expectation Minimization (EM) [68] to reduce manual offline calibration effort.

The likelihood function of the fingerprints is stored either as non-parametric or parametric likelihood model [58]. For non-parametric likelihood model, the

likelihood functions are stored as histograms. For parametric likelihood model, the distribution of the data is assumed to follow a certain distribution function, such as Gaussian distribution. The fingerprint at each location is then modeled as a kernel function with its parameters adjusted according to the characteristics of the data. Like deterministic methods, the most probable location is the location that has the closest match fingerprint to the online fingerprint. This is often obtained by calculating Euclidean distance between the online fingerprint and the statistical mean of the offline fingerprints.

Youssef, Agrawala, and Shankar introduced a joint clustering technique to reduce the computational cost when the size of deployment space grows bigger [69]. The main underlying assumption of this method is that not all access points provide important positioning metric that is unique to user's location. Therefore, the technique selects only a set of access points that are important enough to be included in the calculation of position estimates.

Most methods do not consider small-scale variation which happens when the tracked object's movement is in the order of wavelength. Small-scale variation is more difficult to handle than large-scale variation because the wavelength could be in the order of ten centimeters for gigahertz frequency. Since it is nearly impossible to record fingerprints every few centimeters in a large indoor area, the small-scale variation typically could not be captured in the offline data and therefore the positioning accuracy would be degraded. Based on the assumption that the speed limit of a user is a limiting factor of the physical movement of a

user, the effect of small-scale variation can be managed by applying perturbation technique presented by Youssef and Agrawala [2]. This involves modifying the RSS of some access points by a scaling factor, re-estimating the location, and choosing a location that is closer to the previous location as the new location.

2.5.4 Handling device heterogeneity

Various methods have been proposed to handle device variations, mostly for handling gain variation problem for heterogeneous devices. The most basic technique for handling gain variation is to collect data from each unique device accompanied with information of the device locations manually provided by the users, followed by compensating the fingerprints of each device with a linear first-order model [70]. However, this process is tedious and impractical for large-scale deployment.

Quasi-automatic calibration constructs a gain compensation model solely from the calibration data and the online measurements without requiring users to record the locations at which the calibration data are collected [70] [71]. Quasi-automatic calibration involves searching for the linear parameters that maximizes the confidence level of Markov model [70] or applying weighted least-square optimization [71]. Although quasi-automatic calibration requires less manual work than manual calibration, this approach still need users to differentiate calibration measurements from online measurements. To eliminate the need of

marking the calibration period, automatic calibration has been attempted by applying a simultaneous localization, tracking, and calibration [70]. The accuracy of automatic calibration, however, is still not as good as the accuracy of manual calibration [72].

Calibration-free method is an innovative approach proposed for handling gain variability problem. In calibration-free method, the location fingerprints are stored in a standardized form which is supposedly more independent of gain variation than the non-standardized raw RSS vector that is highly influenced by gain variation. Hyperbolic Location Fingerprinting (HLF) proposed by Kjærgaard and Munk [72] is one of the earliest works that use calibration-free approach. By using RSS ratios of all possible pairs of WLAN access points, HLF produces a more stable positioning result across heterogeneous WLAN clients than RSS. Subsequently, Dong et al. used RSS differences instead of ratios to represent the location fingerprints [73]. Both methods increase the dimension of the fingerprint and therefore increase the computational complexity of the fingerprint comparison. Hossain et al. improved the earlier work by introducing Signal Strength Difference (SSD) [74] that reduces the dimension of the RSS ratios by taking a subset of independent pairs of access points only. The dimensionality reduction, however, is found to produce lower positioning accuracy [25]. Further improvement in calibration-free method is proposed by Zou et al. [57] through the use of a statistical method known as Procrustes analysis to compare the shape similarity between location fingerprints. The shape

similarity is compared by applying linear translation and uniform scaling on each RSS vector. This method, named as Signal Tendency Index (STI), is based on a hypothesis that the shape of a fingerprint is more location-specific than its absolute values. The main advantage of STI is that it does not change the fingerprint dimensionality; therefore, the complexity remains the same as the raw data. Moreover, STI fingerprint tends to produce better positioning accuracy than SSD and RSS [57].

To date, calibration-free methods reported in earlier research works seem to handle gain variation only and assume that the antenna radiation variation does not have any effects on the fingerprints. This requires wireless transceivers to use ideal isotropic antennas or, at least, omnidirectional antennas with the same horizontal/vertical orientation. It has been observed that antenna radiation actually has a big impact on the positioning accuracy [4]. Although it is possible to use omnidirectional antennas for positioning, it is important to note that cheaper and smaller antennas are typically not omnidirectional and thus handling variation in antenna radiation becomes crucial. Nevertheless, we found almost no work that is aimed at handling antenna radiation variation. There appear to be some works that are slightly related to antenna radiation problem, such as the work presented by Dil and Havinga [75], where the fingerprints are collected using two different orientations of omnidirectional antennas, i.e. vertical and horizontal orientation, to train the position estimator. For wearable device, Zhao et al. proposes that indoor positioning systems that use omnidirectional antennas

can benefit from the radiation pattern introduced by the obstruction of human body [76]. This approach opens up the opportunity of simultaneously predicting the device location and human body orientation. Although those two works partly handle antenna radiation problem, the works are still based on omnidirectional antennas, and therefore the antenna radiation variation problems in those two cases are more simplified than if non-omnidirectional antennas are used. A characterization of monopole antennas in multiple RFID devices presented by Yang and Cha [77] shows an extremely irregular antenna radiation variability which poses a highly challenging task in indoor positioning system. This problem is more acute when the antennas being used have unpredictable radiation pattern, such as in low-cost random-wire antennas.

2.5.5 Fusion approach

In fusion approach, position estimation is performed using more than one device or method. This approach can be viewed as an ecosystem where each member performs an optimization process, shares information with other members, and influences the other members [78]. Each member produces a candidate solution that either supports or contradicts candidate solutions produced by other members. When candidate solutions can be consolidated correctly, the accuracy of fusion approach could be higher than the accuracy of non-fusion approach. In other words, the weakness of a member can be compensated by the strength of another member [79].

Inertial or motion sensors such as accelerometer and gyroscope, for instance, are good for recording small movements of moving objects, but they suffer from error accumulation which is known as drifting problem. In contrast, Wi-Fi RSS has positioning errors in meters, which are generally larger than inertial or motion sensors, but it does not encounter any drifting problem. Chen et al. proposed a fusion positioning system comprising inertial sensors and Wi-Fi [80], which are embedded in most smartphones and tablets in the market today. It is found that the cumulative error produced by inertial sensors can be corrected by the position estimate of Wi-Fi fingerprints. On top of that, certain movements which produce unique patterns of inertial sensor data, such as passing through a door or climbing stairs, can be used to identify position landmarks where we can have a high degree of confidence of location prediction and use that prediction to correct the position estimate [81]. An interesting extension of these works is a sensor fusion approach that predicts not only the position of those who wear the device, but also their posture [82] [83]. This is achieved by combining the relative movements obtained from inertial sensors and the coarse position estimates from UWB ToA [82], as well as angular information from a camera [83].

Papaioannou used Wi-Fi RSS to assist camera-based positioning system when the visual data are too ambiguous [84]. On the other hand, unambiguous visual data are used to train Wi-Fi positioning model, eliminating the need of laborious manual data collection to learn radio propagation parameters. A more sophisticated indoor positioning system presented by He, Alooi, and Li [85]

incorporates visual data from external LiDAR-camera as well as mobile phone's Wi-Fi RSS and inertial sensors to build a highly accurate indoor positioning system. This system, however, could have a potential power consumption issue as it uses particle filter method which involves particle updating process that is highly complex.

Current fusion approach tends to focus on achieving higher accuracy and not taking into account other considerations. In fact, positioning algorithm with multiple sensors often requires higher data acquisition rate and higher computational complexity. Fusion approach involving motion sensors requires a high data acquisition rate to avoid inducing large error and, similarly, fusion approach involving visual data requires tracking algorithms with continuous updating process. Those traits are usually not preferred, especially for implementation in portable devices, as the power consumption of the system would be higher and thus reducing battery lifetime. It is therefore worthwhile to explore a more efficient method for consolidating information from various sensors.

Chapter 3

Location Fingerprinting with Continuous Output

3.1 Introduction

Location fingerprinting approach offers a solution to the nonlinear and asymmetric characteristics of indoor radio propagation by remembering the “fingerprints” of positioning metrics at training points. This approach has been developed based on an assumption that positioning metrics measurement of a tracking device has certain unique patterns that can be identified and compared.

Because fingerprint collection often requires tedious manual work, training data are usually collected at discrete and sparsely-distributed points. Meanwhile, the locations of the tracking device during testing phase are uncertain and may not be located at the exact coordinates of the training points.

Based on the type of its output space, location fingerprinting methods can either have continuous output space or discrete output space. As the name implies, continuous-output methods have continuous location estimates over the physical space, whereas discrete-output methods have a restricted set of location estimates in physical space as illustrated in Figure 3.1. Typically, the set of location estimates in discrete-output methods is either the location of reference devices or the location of training points.

Figure 3.1: Continuous and discrete output space in location fingerprinting
(Legend: o = output, x = location of reference devices or training points)

Existing location fingerprinting methods with continuous output offer better precision but lack robustness against noisy data. In contrast, discrete output tends to be more robust to noise, but its precision is directly limited by the location of the anchor or reference devices.

In the first part of this chapter, several existing continuous-output and discrete-output location fingerprinting approaches are presented. Then, we present our proposed approach, Continuous-Output Location Estimation using Model Tree (CO-LEMT), combining the principle of continuous and discrete location fingerprinting approach to achieve indoor positioning approach with better accuracy.

3.2 Location fingerprinting

Consider a location fingerprinting system that consists of m reference devices, one tracking device, and p anchors. All anchors and reference devices are placed at fixed locations which are known to the system. The tracking device can be placed at one of n possible locations in set $\mathbf{L} = \{l_1, l_2, \dots, l_n\}$. The 2-dimensional coordinate of the i -th possible physical location is $l_i = (x_i, y_i)$ where $l_i \in \mathbf{L}$.

We define training points and testing points as sets of locations of the tracking device during training phase and testing phase, respectively. The coordinate of the tracking device is known to the indoor positioning system during training phase, but not during testing phase. To simplify the problem, we assume that both training points and testing points are subset of $\mathbf{L}$.

Reference devices, on the other hand, can be placed at any coordinates outside the set $\mathbf{L}$. Therefore, they do not have location index as the tracking device does.

For each i -th location, the RSS of the k -th reference device $r_{kj}^{(i)}$ and the RSS of the tracking device $s_j^{(i)}$ are measured by the j -th anchor during training phase. In testing phase, the RSS of the k -th reference device and tracking device measured by the j -th anchor are denoted as $\bar{r}_{kj}$ and $\bar{s}_j$ respectively. The RSS vector of all reference devices during training phase and testing phase are defined as $R_j^{(i)} = \{r_{1j}^{(i)}, r_{2j}^{(i)}, \dots, r_{mj}^{(i)}\}$ and $\bar{R}_j^{(i)} = \{\bar{r}_{1j}^{(i)}, \bar{r}_{2j}^{(i)}, \dots, \bar{r}_{mj}^{(i)}\}$, respectively. Mathematical

symbols and meanings defined above are used throughout the rest of this chapter, unless explicitly stated otherwise.

The types of information used in location fingerprinting techniques are as summarized in Table 3.1, which consist of RSS of tags and the actual location of tags associated with reference and tracking device during training phase and testing phase.

Table 3.1: Types of information used in location fingerprinting system

	Reference tags	Tracking tag
Training phase	<ul style="list-style-type: none"> • RSS • Location coordinate 	<ul style="list-style-type: none"> • RSS • Location coordinate • Location index
Testing phase	<ul style="list-style-type: none"> • RSS • Location coordinate 	<ul style="list-style-type: none"> • RSS

The output of the location fingerprinting can either be the location index or the location coordinate of the tracking tag. While it is easy to map a discrete location index into a continuous location coordinate, the other way around may not be as straightforward because the coordinate might not be included in the set of possible physical locations. Therefore, we choose location coordinates as the end results of the fingerprinting techniques.

3.2.1 Nearest neighbor

Nearest Neighbor (NN) [4] is developed based on an observation that a tracking device at a certain physical location has a unique RSS fingerprints. If the environmental condition is static, fingerprints can be captured and stored during training phase, and later compared with the fingerprints obtained during testing phase. The location of the nearest neighbor in RSS space, which is the stored fingerprint that has the shortest fingerprint distance to the testing fingerprint, most likely has the shortest distance to the testing location in physical space.

Euclidean distance and the selection of the nearest neighbor are formulated as:

$$D_i = \sqrt{\sum_{j=1}^p \left(s_j^{(i)} - \bar{s}_j \right)^2} \quad (3.1)$$

$$L = \arg \min_{l_i \in L} D_i \quad (3.2)$$

As indicated by Eq. (3.2), NN method produces a discrete output where the output of the method is limited to the set of training points. Testing point is always identified as one of the training points as shown in the illustration in Figure 3.2. As a result, the precision of location estimation would decrease when training points are more sparsely distributed.

Figure 3.2: Sparse training points in discrete-output nearest neighbor

It is also possible to involve multiple nearest neighbors instead of just one nearest neighbor. However, since improvement from using multiple nearest neighbors is found to be insignificant [4], we only consider single nearest neighbor and not the multiple nearest neighbors technique in this work.

Besides having limited precision due to its discrete-output nature, NN requires repeated collections of fingerprints to handle dynamic changes in the environmental condition. If the RSS fingerprints collected during training phase do not represent fingerprints in the testing phase, the fingerprints need to be recollected and updated. Otherwise, the accuracy would degrade in the event of temporal variation.

3.2.2 LANDMARC

LANDMARC learns the environmental condition in an online manner by using reference tags placed at fixed position to assist the capture of RSS [10]. The

coordinate of the tracking device is inferred from the coordinates of the reference devices and the RSS of the nearest reference devices. This technique assumes that the RSS fingerprint of the tracking device at a certain location is correlated in some way with the RSS fingerprints of the reference devices that are physically close to the tracking device.

For $k = 1, \dots, m$, Euclidean distance between the fingerprint of the tracking device and the fingerprint of each reference device are calculated as:

$$D_k = \sqrt{\sum_{j=1}^p |\bar{r}_{kj} - \bar{s}_j|^2} \quad (3.3)$$

The Euclidean distances $D = \{D_1, D_2, \dots, D_m\}$ is sorted in ascending manner to form $\dot{D} = \{\dot{D}_1, \dot{D}_2, \dots, \dot{D}_m\}$, where $\dot{D}_1$ is the smallest distance and $\dot{D}_m$ is the largest distance.

The k -smallest Euclidean distances are then used to calculate the weights:

$$w_u = \frac{\frac{1}{\dot{D}_u^2}}{\sum_{u=1}^k \frac{1}{\dot{D}_u^2}} \quad (3.4)$$

where u is the index of the corresponding distance measure that has been sorted in ascending manner. Ni et al. found that the use of $k = 4$ empirically produce the best accuracy [10].

The estimated coordinate of the tracking tag, $(\hat{x}, \hat{y})$, is then computed as:

$$(\hat{x}, \hat{y}) = \sum_{u=1}^k w_u (x_u, y_u) \quad (3.5)$$

where (x_u, y_u) is the coordinate of reference tag u .

LANDMARC provides a convenient way to automatically adapt to the current environmental condition, particularly because it does not require any manual data collection as in the training phase of nearest neighbor technique.

However, despite its advantages, there are several known limitations of LANDMARC method. First, all devices involved must be calibrated beforehand to avoid inaccuracy caused by device variation. It is also found that the accuracy of the system is significantly affected by the density of the reference devices. Besides, when the propagation paths between the tracking device and the nearest reference devices are unbalanced, for example when one of the nearest reference devices is blocked by wall but the rest are not, the accuracy could be significantly affected.

3.2.3 LEMT

Similar to LANDMARC, Location Estimation using Model Tree (LEMT) [6] relies on reference devices to deal with dynamic environment. However, LEMT is found to produce better accuracy than LANDMARC when using less number of reference devices [5].

LEMT considers that the measured RSS may change over time. However, it assumes that the functional relationship between devices does not change over time and can be learned from training data. The functional relationship between the reference devices and the tracking device, $f_{ij}(\cdot)$, that is learned during training phase is defined as:

$$s_j^{(i)} = f_{ij}(r_{1j}^{(i)}, r_{2j}^{(i)}, \dots, r_{mj}^{(i)}) \quad (3.6)$$

Compared to NN and LANDMARC, LEMT requires much longer time since its computation is much more complex. NN only involves a simple record-and-compare mechanism of the fingerprints, whereas LANDMARC applies simple functions that directly estimate the current position of the tracking device. In LEMT, a complex functional relationship model is used to estimate the RSS of the tracking device if the tracking device were located at the known training coordinates, which are then used to estimate the position of the tracking device.

Since RSS vectors of tracking device and reference devices are multidimensional, manually analyzing the functional relationship model is challenging and not a feasible option. This is an area where machine learning could be of help. LEMT uses M5 model tree proposed by Quinlan [86] to model the functional relationship, $f_{ij}(\cdot)$. During training phase, a decision tree containing several piecewise-linear functions in M5 model tree is constructed.

M5 model tree initially builds the decision tree by recursively splitting the training data into a structure of interior nodes and leaf nodes, with the objective

of obtaining the maximum reduction in error. Then, the tree is pruned by examining subtrees that can be converted into leaf nodes. After pruning, each leaf node is assigned with a linear model, resulting in a decision tree that contains multiple linear models on its leaves as illustrated in Figure 3.3.

Figure 3.3: An example of M5 model tree representing functional relationship between RSS of a tracking device, $s_j^{(i)}$, and RSS of reference devices, $r_{kj}^{(i)}$

Given the RSS of all reference devices and the RSS of tracking device measured by all anchors during testing phase, the objective of LEMT is to find L , the most likely location of the tracking device, where $L \in \mathbf{L}$. Assuming functional relationship between the reference devices and tracking device does not change over time, M5 model tree predicts the RSS of tracking device $\sigma_j^{(i)}$ from the RSS vector of reference devices $(\bar{r}_{1j}, \bar{r}_{2j}, \dots, \bar{r}_{mj})$, if the tracking device is assumed at

location i , by using the functional relationship that has been learned during training phase:

$$\sigma_j^{(i)} = f_{ij}(\bar{r}_{1j}, \bar{r}_{2j}, \dots, \bar{r}_{mj}) \quad (3.7)$$

The Euclidean distance between the predicted value and the true RSS of the tracking device at location i is then calculated as:

$$D_i = \sqrt{\sum_{j=1}^p (\sigma_j^{(i)} - \bar{s}_j)^2} \quad (3.8)$$

The most likely location of the tracking device during testing phase is the one that minimizes the Euclidean distance, as defined by:

$$L = \arg \min_{l_i \in L} D_i \quad (3.9)$$

The time complexity of LEMT algorithm is $\mathcal{O}(m'np)$, where m' is the average depth of the model tree, n is the number of possible physical locations, and p is the number of anchors. Therefore, the training time and testing time required to perform LEMT are linearly affected by the number of devices and the locations to distinguish.

3.3 Methodology of CO-LEMT

As we present in Table 3.2, existing fingerprinting techniques use different sets of input and different types of output. NN uses fingerprints collected from training points only, LANDMARC uses fingerprints collected from reference devices only, whereas LEMT uses both fingerprints collected from training points and reference devices.

Table 3.2: Comparison of location fingerprint input, output, and weakness

Method name	Input		Output	Weakness
	Reference devices	Training points		
NN	Not used	Used	Discrete	<ul style="list-style-type: none">• Dense training points required• Susceptible to temporal variation
LANDMARC	Used	Not used	Continuous	<ul style="list-style-type: none">• Dense reference devices required
LEMT	Used	Used	Discrete	<ul style="list-style-type: none">• Dense training points required• Long training time

Each technique has its own weaknesses. NN needs the training points to be dense enough to achieve good accuracy. This is highly discouraged as the collection of fingerprints at training points typically requires tedious manual work where every point is to be visited during training phase. Moreover, the fingerprints must be recaptured from time to time to anticipate the changing fingerprints in a dynamic environment. In LANDMARC, the use of reference devices allows us to learn in a dynamic environment automatically, but it would contribute to a higher deployment cost especially when a high number of reference devices is required.

Yin, Yang, and Ni found that LEMT produces better accuracy than LANDMARC with less number of reference devices and anchor devices [6]. Since the number of devices is directly related to cost, the ability to use less number of devices allows the positioning system to be economically deployed in large scale systems. The study however, does not consider the number of training points which is related to the manpower needed to collect the training data. This factor is critical because the highest precision of discrete-output location fingerprinting is limited by the density of the training points.

To address this problem, we propose a new location fingerprinting method called Continuous-Output LEMT (CO-LEMT) that combines the concept of LEMT and LANDMARC as presented in Figure 3.4.

Figure 3.4: Block diagram of CO-LEMT

To develop CO-LEMT, we consider the basic idea of LEMT represented in Eq. (3.7), which assumes that the RSS of the tracking device and the RSS of reference devices have a functional relationship that can be learned from the training data. Once the functional relationship has been learned, the estimated RSS of the tracking device at each training point, $\sigma_j^{(i)}$, which we refer as virtual reference points, is continuously updated according to the real-time RSS data produced by reference devices during testing phase.

In the original LEMT method, the chosen fingerprint is the one that has the minimum Euclidean distance, producing a discrete output that is bounded by the locations of training points. To produce a continuous output, instead of directly choosing a virtual reference point that has the minimum Euclidean distance, we adopt k -nearest-neighbors method used in LANDMARC with the exception that the reference fingerprints are taken from the RSS vectors at virtual reference points, such that Eq. (3.3) is modified as follow:

$$D_i = \sqrt{\sum_{j=1}^p |\sigma_j^{(i)} - \bar{s}_j|^2} \quad (3.10)$$

Once the distance values have been computed, we choose k neighbors that have the smallest distances, apply weight calculation, and calculate the location estimate by following through the remaining computation in LANDMARC algorithm as described in Eq. (3.4) and Eq. (3.5).

The proposed CO-LEMT algorithm is as summarized in Algorithm 1.

Algorithm 1: Summary of CO-LEMT

Pre-processing

1. Compute the average of RSS of reference devices and tracking device during training phase and testing phase for every T_{avg} seconds.
2. If there are missing data in a reference device or tracking device, impute the missing data with the lowest possible value of RSS.

CO-LEMT algorithm

1. Training phase:
For each anchor j at each i -th location, train functional relationship model $f_{ij}(\cdot)$ to fit the training data. The model is based on M5 model tree, whereas the training data consists of the averaged RSS vector of reference devices as training input, $r_{kj}^{(i)}$, and the averaged RSS data of tracking device as training target, $s_j^{(i)}$.
2. Testing phase:
 - a. For each anchor j at each i -th location of training points, feed in the average RSS vector of reference devices captured during testing phase $\bar{R}_{kj}$ into the functional relationship model, $f_{ij}(\cdot)$. The output of the model is the RSS estimate of the tracking device at the training points, $\sigma_j^{(i)}$.
 - b. Compute the Euclidean distances between the training fingerprints at virtual reference points and the testing fingerprint.

$$D_i = \sqrt{\sum_{j=1}^p |\sigma_j^{(i)} - \bar{s}_j|^2}$$

- c. Sort the Euclidean distance from the smallest to the largest and use the k smallest distance to compute the weight.

$$w_u = \frac{\frac{1}{D_u^2}}{\sum_{u=1}^k \frac{1}{D_u^2}}$$

- d. Compute the estimated coordinate of the tracking device.

$$(\hat{x}, \hat{y}) = \sum_{u=1}^k w_u (x_u, y_u)$$

3.4 Performance evaluation

3.4.1 Experimental setup

Research works in indoor positioning involve data that are produced from either simulation or real equipment. Simulation data are easier to produce and offer more flexibility to vary the test parameters. However, the generation of simulation data relies on various assumptions and thus the characteristics of indoor environment may not be represented accurately. As opposed to simulation data, real data are more difficult to produce, but the characteristics of the indoor environment can be captured as they are. To obtain more realistic results, we performed our experiments using real data instead of simulation data. We collected the data using 2.4-GHz active RFID system containing a network of readers and tags. One of the readers acts as a coordinator that consolidates all the data obtained by all readers and forwards them to a computer via a serial cable.

Each tag periodically broadcasts ping signal to the surrounding readers. When a reader senses a ping signal, it measures the RSS and reports it to the coordinator. Signal from each active RFID tag can reach readers within a radius of 5 m to 20 m, depending on the layout of the building and the orientation of the devices.

Throughout the experiment, RFID tags were configured to send ping signals every 3 seconds. The tracking device was placed at the collection points for 1 minute each. Reference devices are placed at static locations on the test site in addition to anchor devices and tracking device.

For each dataset, we conducted 3 data collection campaigns. Between one campaign and its subsequent campaign, there is at least a 1-hour break. From the 3 collection campaigns for each dataset, we arrange 2 different cases to simulate the condition: 1) under spatial variation, and 2) under spatial-temporal variation. In the first case, training data and testing data are drawn from the same campaign. In the latter case, they are drawn from different campaigns, of which we applied permutation to decide which campaign indices are used for training and testing.

Training data are used to adjust the parameters of the propagation model that maps the RSS into distance estimation. Meanwhile, testing data are used to find the location estimate which will then be evaluated in terms of accuracy and processing time. The accuracy is represented as 90th and 95th error percentile, which is the distance error in meters where 90% and 95% of the location estimates fall under the value.

We collected RSS data at several locations with different setups and different number of devices. Dataset A and dataset C were collected at the same test-site. In dataset A, an RFID tag is used as a tracking device, the readers are treated as anchors, and the remaining tags are treated as reference devices. The roles of devices are switched in dataset C, where an RFID reader is used as a tracking device, tags are treated as anchors, and the remaining readers are treated as reference devices. Dataset B was collected using a device configuration like dataset A at another test-site.

Figure 3.5 shows the site layouts used to produce each dataset. For scenarios that use less points or devices, we selected corner locations since the sites are relatively small, thus almost all reference and tracking devices are still within the range of the anchors. With this configuration, the corner locations are more likely to capture extreme conditions of the RSS fingerprints and to represent the overall fingerprints better. For larger-scale testing, the corner locations may not be adequate to represent the fingerprints and distributed locations might be required.

To evaluate the robustness of our proposed method under various conditions, we generate 2 test cases involving spatial variation and spatial-temporal variation, and 4 types of scenarios where the number of reference devices, anchors, and training points are varied as summarized in Table 3.3.

Table 3.3: Datasets and test scenarios for location fingerprinting

		Dataset								
		A			B			C		
Test area		6 m x 10 m			12 m x 3 m			6 m x 10 m		
Device configuration		Readers as anchors			Readers as anchors			Tags as anchors		
		# ref	# anchor	# training	# ref	# anchor	# training	# ref	# anchor	# training
Scenario	Complete set	10	5	20	7	4	10	5	10	20
	Less reference	4	5	20	4	4	10	3	10	20
	Less anchor	10	3	20	7	2	10	5	4	20
	Less training	10	5	3	7	4	4	5	10	3

(a) Layout for dataset A

(b) Layout for dataset B

(c) Layout for dataset C

Figure 3.5: Site layout for performance evaluation of location fingerprinting

All performance evaluation presented in this thesis were conducted using Matlab 2017a running on Windows 10 machine with dual-core Intel Core with 8 GB physical memory. We performed 10 trials for each testing and recorded the accuracy and time in terms of mean and standard deviation.

It should be noted that the use of “ $\pm$ ” symbol in performance tables presented in this thesis is solely meant to separate standard deviation value from the mean value, not to imply that the results would follow a certain distribution. We also present cumulative distribution plots, which capture the distribution of error obtained from one trial. Because the plots capture the distribution of error for one trial, these plots may not have the same exact values of the 90th and 95th error percentiles represented in the performance tables which consolidate the results obtained from 10 trials and a permutation of campaign indices.

3.4.2 Performance result under spatial variation

As shown in Table 3.4, the proposed CO-LEMT produces the lowest error than the benchmark methods under spatial variation. The error of CO-LEMT is substantially lower than NN and LANDMARC and it is slightly lower than the error of LEMT, except for dataset B where LEMT method sometimes achieves zero errors. These zero errors, however, is an indication of overfitting. We verify this in the next experiment under spatial-temporal variation.

Table 3.4: Accuracy of location fingerprinting under spatial variation

Dataset - Scenario	Method			
	NN	LANDMARC	LEMT	Proposed CO-LEMT
	90th error percentile in meters (mean $\pm$ standard deviation)			
A – Complete set	2.40 $\pm$ 0.38	2.59 $\pm$ 0.22	1.63 $\pm$ 0.13	1.29 $\pm$ 0.05
B – Complete set	1.20 $\pm$ 1.73	4.42 $\pm$ 0.40	0.00 $\pm$ 0.00	0.26 $\pm$ 0.31
C – Complete set	2.03 $\pm$ 0.01	2.76 $\pm$ 0.11	1.49 $\pm$ 0.22	1.25 $\pm$ 0.09
A – Less reference	2.40 $\pm$ 0.38	2.91 $\pm$ 0.27	1.77 $\pm$ 0.12	1.57 $\pm$ 0.10
B – Less reference	1.20 $\pm$ 1.73	6.76 $\pm$ 0.45	0.67 $\pm$ 0.96	0.65 $\pm$ 0.81
C – Less reference	2.03 $\pm$ 0.01	2.82 $\pm$ 0.11	1.56 $\pm$ 0.14	1.30 $\pm$ 0.06
A – Less anchor	2.78 $\pm$ 0.29	2.99 $\pm$ 0.13	1.70 $\pm$ 0.00	1.54 $\pm$ 0.13
B – Less anchor	1.60 $\pm$ 2.30	4.59 $\pm$ 0.64	0.00 $\pm$ 0.00	0.18 $\pm$ 0.12
C – Less anchor	2.61 $\pm$ 0.29	2.78 $\pm$ 0.03	1.91 $\pm$ 0.12	1.69 $\pm$ 0.08
A – Less training	3.82 $\pm$ 0.41	2.59 $\pm$ 0.22	3.89 $\pm$ 0.39	2.84 $\pm$ 0.26
B – Less training	5.01 $\pm$ 1.33	4.42 $\pm$ 0.40	3.47 $\pm$ 1.18	4.03 $\pm$ 0.36
C – Less training	3.57 $\pm$ 0.05	2.76 $\pm$ 0.11	3.86 $\pm$ 0.44	2.63 $\pm$ 0.13
	95th error percentile in meters (mean $\pm$ standard deviation)			
A – Complete set	3.01 $\pm$ 0.33	3.12 $\pm$ 0.14	2.01 $\pm$ 0.02	1.69 $\pm$ 0.10
B – Complete set	2.20 $\pm$ 3.16	4.98 $\pm$ 0.00	0.00 $\pm$ 0.00	0.50 $\pm$ 0.51
C – Complete set	2.91 $\pm$ 0.19	3.13 $\pm$ 0.10	1.91 $\pm$ 0.12	1.67 $\pm$ 0.08
A – Less reference	3.01 $\pm$ 0.33	3.40 $\pm$ 0.20	2.27 $\pm$ 0.29	1.91 $\pm$ 0.15
B – Less reference	2.20 $\pm$ 3.16	7.72 $\pm$ 0.24	1.20 $\pm$ 1.73	1.30 $\pm$ 1.68
C – Less reference	2.91 $\pm$ 0.19	3.26 $\pm$ 0.15	1.95 $\pm$ 0.15	1.63 $\pm$ 0.10
A – Less anchor	3.56 $\pm$ 0.62	3.56 $\pm$ 0.25	2.38 $\pm$ 0.40	1.98 $\pm$ 0.04
B – Less anchor	2.26 $\pm$ 2.00	5.26 $\pm$ 0.51	0.41 $\pm$ 0.59	0.58 $\pm$ 0.43
C – Less anchor	3.45 $\pm$ 0.54	3.20 $\pm$ 0.16	2.67 $\pm$ 0.20	2.06 $\pm$ 0.04
A – Less training	4.84 $\pm$ 0.44	3.12 $\pm$ 0.14	5.06 $\pm$ 0.56	3.33 $\pm$ 0.34
B – Less training	6.08 $\pm$ 1.69	4.81 $\pm$ 0.19	6.08 $\pm$ 1.69	5.08 $\pm$ 0.69
C – Less training	4.76 $\pm$ 0.86	3.13 $\pm$ 0.10	4.97 $\pm$ 0.30	3.03 $\pm$ 0.13

(a) Dataset A – Complete set

(b) Dataset A – Less reference

(c) Dataset A – Less anchor

(d) Dataset A – Less training

Figure 3.6: Cumulative error distribution of location fingerprinting for Dataset A under spatial variation

Cumulative distribution plots in Figure 3.6, Figure 3.7, and Figure 3.8, show that the proposed CO-LEMT errors for scenarios with complete set, less reference, and less anchor are comparable with LEMT errors, which is better than NN and LANDMARC. However, in the case of less training, LEMT error generally becomes much bigger, while the proposed error tends to be similar with the LANDMARC error. Those results indicate that the proposed CO-LEMT, which combines the LEMT and LANDMARC methodology, produce performance results that combines the advantages of both methods on different scenarios. This

is expected since CO-LEMT uses both training data and reference data.

Reference data could remedy the lack of training data, and vice versa.

(a) Dataset B – Complete set

(b) Dataset B – Less reference

(c) Dataset B – Less anchor

(d) Dataset B – Less training

Figure 3.7: Cumulative error distribution of location fingerprinting for Dataset B under spatial variation

However, in Table 3.4, there is an anomaly on Dataset B for the scenario with less training points, where the 90th percentile of LEMT is the lowest among other methods, but its 95th percentile is higher. This anomaly can be seen more clearly in Figure 3.7, where the LEMT error is smaller than other methods for the 90th error percentile. One possible explanation of this anomaly is that, as described in Table 3.3, dataset B has the highest number of training points for the scenario

with less training points. The scenario with less training points for dataset B uses 4 out of 10 training points in the complete set. It is not as low as the other two datasets, where there are only 3 out of 20 training points in the complete set used in the scenario with less training points. Therefore, it is possible that training points in dataset B still are still enough to represent the environment, resulting in good LEMT accuracy results for this dataset. However, if we reduce the number of training points further, there could be an accuracy degradation on LEMT, and the proposed CO-LEMT might produce better accuracy in this case.

(a) Dataset C – Complete set

(b) Dataset C – Less reference

(c) Dataset C – Less anchor

(d) Dataset C – Less training

Figure 3.8: Cumulative error distribution of location fingerprinting for Dataset C under spatial variation

In Table 3.5, we show the training time and testing time comparison of the methods. It should be noted that NN and LANDMARC do not require training.

Table 3.5: Training time and testing time of location fingerprinting under spatial variation

Dataset - Scenario	Method			
	NN	LANDMARC	LEMT	Proposed CO-LEMT
	Training time in milliseconds (mean $\pm$ standard deviation)			
A – Complete set	-	-	912.25 $\pm$ 189.86	921.86 $\pm$ 214.48
B – Complete set	-	-	323.63 $\pm$ 39.40	324.54 $\pm$ 42.89
C – Complete set	-	-	1093.96 $\pm$ 106.50	1078.43 $\pm$ 106.49
A – Less reference	-	-	462.69 $\pm$ 44.16	456.75 $\pm$ 34.64
B – Less reference	-	-	229.28 $\pm$ 32.39	226.74 $\pm$ 35.23
C – Less reference	-	-	662.67 $\pm$ 79.72	660.31 $\pm$ 79.43
A – Less anchor	-	-	431.87 $\pm$ 26.54	434.60 $\pm$ 26.41
B – Less anchor	-	-	164.46 $\pm$ 20.83	159.10 $\pm$ 20.17
C – Less anchor	-	-	396.53 $\pm$ 35.33	399.28 $\pm$ 35.10
A – Less training	-	-	104.26 $\pm$ 12.51	104.83 $\pm$ 13.46
B – Less training	-	-	129.87 $\pm$ 18.58	124.25 $\pm$ 5.99
C – Less training	-	-	121.63 $\pm$ 21.32	120.75 $\pm$ 19.86
	Testing time in milliseconds (mean $\pm$ standard deviation)			
A – Complete set	2.56 $\pm$ 0.48	7.54 $\pm$ 1.35	623.41 $\pm$ 138.36	595.38 $\pm$ 102.91
B – Complete set	1.31 $\pm$ 0.32	3.58 $\pm$ 0.72	102.80 $\pm$ 14.21	102.46 $\pm$ 20.38
C – Complete set	2.63 $\pm$ 0.05	5.69 $\pm$ 0.15	1078.54 $\pm$ 90.78	1056.52 $\pm$ 82.73
A – Less reference	2.27 $\pm$ 0.32	4.96 $\pm$ 0.72	467.74 $\pm$ 55.21	464.46 $\pm$ 64.56
B – Less reference	1.23 $\pm$ 0.24	2.82 $\pm$ 0.43	93.41 $\pm$ 9.80	93.19 $\pm$ 9.34
C – Less reference	2.40 $\pm$ 0.07	4.59 $\pm$ 0.24	797.00 $\pm$ 39.00	796.18 $\pm$ 44.54
A – Less anchor	2.15 $\pm$ 0.21	6.43 $\pm$ 0.37	288.26 $\pm$ 12.11	280.48 $\pm$ 23.97
B – Less anchor	1.23 $\pm$ 0.33	3.53 $\pm$ 1.19	54.28 $\pm$ 7.13	54.60 $\pm$ 6.98
C – Less anchor	2.16 $\pm$ 0.09	5.12 $\pm$ 0.52	365.96 $\pm$ 20.51	354.33 $\pm$ 12.95
A – Less training	2.04 $\pm$ 0.19	6.60 $\pm$ 0.73	69.72 $\pm$ 3.40	72.67 $\pm$ 3.28
B – Less training	1.07 $\pm$ 0.21	3.03 $\pm$ 0.48	39.15 $\pm$ 4.45	40.72 $\pm$ 3.54
C – Less training	2.07 $\pm$ 0.09	5.15 $\pm$ 0.19	121.63 $\pm$ 11.91	123.88 $\pm$ 11.37

In terms of training time and testing time, CO-LEMT requires about the same time as LEMT, but hundreds of times slower than NN and LANDMARC for any dataset and any scenario. Similarity between CO-LEMT and LEMT processing time is reasonable since CO-LEMT uses LEMT as its underlying method besides involving LANDMARC as an additional step to obtain the estimated locations. Since the processing time of LEMT is significantly long, the processing time of LANDMARC computation in the CO-LEMT becomes negligible and does not affect the overall processing time.

The longest processing time for LEMT and CO-LEMT is about 1 s for testing time and 1 s for training time, while both NN and LANDMARC only require less than 6 ms testing time without any training time for the same dataset. The CO-LEMT issue with long processing time will be addressed in Chapter 4.

3.4.3 Performance result under spatial-temporal variation

Similar with the result under spatial variation, the spatial-temporal result presented in Table 3.6 shows that the proposed CO-LEMT generally has better accuracy than the benchmark methods under spatial-temporal variation.

All methods have higher error percentiles under spatial-temporal variation than spatial variation. As expected, temporal variation affects the accuracy of location fingerprinting methods.

Table 3.6: Accuracy of location fingerprinting under spatial-temporal variation

Dataset - Scenario	Method			
	NN	LANDMARC	LEMT	Proposed CO-LEMT
	90th error percentile in meters (mean $\pm$ standard deviation)			
A – Complete set	2.92 $\pm$ 0.23	2.60 $\pm$ 0.23	2.98 $\pm$ 0.30	2.36 $\pm$ 0.16
B – Complete set	3.99 $\pm$ 1.63	4.42 $\pm$ 0.40	3.82 $\pm$ 2.82	3.67 $\pm$ 0.53
C – Complete set	2.74 $\pm$ 0.29	2.76 $\pm$ 0.11	2.82 $\pm$ 0.29	2.10 $\pm$ 0.23
A – Less reference	2.92 $\pm$ 0.23	2.93 $\pm$ 0.28	3.02 $\pm$ 0.26	2.38 $\pm$ 0.23
B – Less reference	3.99 $\pm$ 1.63	6.76 $\pm$ 0.45	3.73 $\pm$ 2.31	3.67 $\pm$ 0.54
C – Less reference	2.74 $\pm$ 0.29	2.83 $\pm$ 0.11	2.74 $\pm$ 0.35	2.03 $\pm$ 0.24
A – Less anchor	3.20 $\pm$ 0.26	2.99 $\pm$ 0.13	3.24 $\pm$ 0.34	2.60 $\pm$ 0.23
B – Less anchor	3.56 $\pm$ 1.07	4.59 $\pm$ 0.63	3.93 $\pm$ 0.77	3.36 $\pm$ 0.69
C – Less anchor	2.93 $\pm$ 0.21	2.78 $\pm$ 0.03	3.12 $\pm$ 0.25	2.46 $\pm$ 0.17
A – Less training	3.92 $\pm$ 0.51	2.60 $\pm$ 0.23	4.34 $\pm$ 0.59	2.95 $\pm$ 0.39
B – Less training	4.84 $\pm$ 2.09	4.42 $\pm$ 0.40	5.46 $\pm$ 0.82	4.10 $\pm$ 0.24
C – Less training	4.11 $\pm$ 0.57	2.76 $\pm$ 0.11	3.90 $\pm$ 0.84	2.85 $\pm$ 0.60
	95th error percentile in meters (mean $\pm$ standard deviation)			
A – Complete set	3.36 $\pm$ 0.28	3.13 $\pm$ 0.14	3.54 $\pm$ 0.43	2.86 $\pm$ 0.20
B – Complete set	5.69 $\pm$ 2.18	4.81 $\pm$ 0.19	4.81 $\pm$ 2.72	4.54 $\pm$ 0.62
C – Complete set	3.23 $\pm$ 0.26	3.13 $\pm$ 0.10	3.34 $\pm$ 0.55	2.81 $\pm$ 0.45
A – Less reference	3.36 $\pm$ 0.28	3.42 $\pm$ 0.20	3.51 $\pm$ 0.31	2.91 $\pm$ 0.14
B – Less reference	5.69 $\pm$ 2.18	7.72 $\pm$ 0.24	5.57 $\pm$ 0.89	4.25 $\pm$ 0.48
C – Less reference	3.23 $\pm$ 0.26	3.27 $\pm$ 0.16	3.21 $\pm$ 0.43	2.69 $\pm$ 0.38
A – Less anchor	3.84 $\pm$ 0.40	3.57 $\pm$ 0.25	3.83 $\pm$ 0.41	3.01 $\pm$ 0.22
B – Less anchor	4.34 $\pm$ 1.27	5.26 $\pm$ 0.50	4.47 $\pm$ 0.62	3.66 $\pm$ 0.92
C – Less anchor	3.65 $\pm$ 0.46	3.21 $\pm$ 0.16	3.74 $\pm$ 0.32	2.96 $\pm$ 0.17
A – Less training	5.04 $\pm$ 0.78	3.13 $\pm$ 0.14	5.26 $\pm$ 0.75	3.38 $\pm$ 0.41
B – Less training	5.28 $\pm$ 2.54	4.81 $\pm$ 0.19	6.10 $\pm$ 0.89	4.51 $\pm$ 0.42
C – Less training	5.32 $\pm$ 0.84	3.13 $\pm$ 0.10	5.01 $\pm$ 0.92	3.27 $\pm$ 0.71

Unlike in spatial variation case, LEMT does not produce zero error under spatial-temporal variation case. This confirms that the LEMT's zero error under spatial variation is caused by overfitting and is not an indication of better accuracy. For dataset B that has a strong indication of overfitting on LEMT, our proposed CO-LEMT also experiences an increase of error, but it still has the lowest error among the tested methods.

Figure 3.9: Cumulative error distribution of location fingerprinting for Dataset A under spatial-temporal variation

In Figure 3.9, Figure 3.10, and Figure 3.11, we could see that the results under spatial-temporal variation are relatively similar with results under spatial

variation, except for Dataset A with less training points in Figure 3.9 (d) where LANDMARC produces better accuracy than other methods. Since LANDMARC only considers reference data from static reference devices, it might sometimes work better than other location fingerprinting methods when temporal variation is dominant. However, since CO-LEMT remains the one with the lowest 90th and 95th error percentile in most scenarios, we consider LANDMARC performance for Dataset A with less training points as an anomaly.

(a) Dataset B – Complete set

(b) Dataset B – Less reference

(c) Dataset B – Less anchor

(d) Dataset B – Less training

Figure 3.10: Cumulative error distribution of location fingerprinting for Dataset B under spatial-temporal variation

(a) Dataset C – Complete set

(b) Dataset C – Less reference

Figure 3.11: Cumulative error distribution of location fingerprinting for Dataset C under spatial-temporal variation

In terms of processing time, from Table 3.7 we can observe that the training time and testing time of CO-LEMT are about the same as LEMT. There is a minor increase in the testing time of CO-LEMT when compared to LEMT, which is caused by the additional computation in CO-LEMT to obtain the estimated locations. The testing time of both CO-LEMT and LEMT are much higher than NN and LANDMARC. Those results are consistent with the results we obtained under spatial variation.

Table 3.7: Training time and testing time of location fingerprinting
under spatial-temporal variation

Dataset - Scenario	Method			
	NN	LANDMARC	LEMT	Proposed CO-LEMT
	Training time in milliseconds (mean $\pm$ standard deviation)			
A – Complete set	-	-	1006.54 $\pm$ 135.75	995.46 $\pm$ 122.64
B – Complete set	-	-	323.87 $\pm$ 45.14	322.71 $\pm$ 44.51
C – Complete set	-	-	1300.93 $\pm$ 180.92	1282.28 $\pm$ 183.58
A – Less reference	-	-	544.01 $\pm$ 62.03	541.70 $\pm$ 57.23
B – Less reference	-	-	211.16 $\pm$ 30.80	210.86 $\pm$ 29.13
C – Less reference	-	-	871.86 $\pm$ 107.93	844.00 $\pm$ 109.85
A – Less anchor	-	-	688.85 $\pm$ 218.93	673.60 $\pm$ 208.97
B – Less anchor	-	-	173.65 $\pm$ 43.14	174.58 $\pm$ 38.11
C – Less anchor	-	-	889.23 $\pm$ 103.68	888.21 $\pm$ 100.69
A – Less training	-	-	250.99 $\pm$ 23.46	245.39 $\pm$ 31.04
B – Less training	-	-	238.96 $\pm$ 26.99	235.23 $\pm$ 19.37
C – Less training	-	-	291.38 $\pm$ 28.81	298.52 $\pm$ 40.71
	Testing time in milliseconds (mean $\pm$ standard deviation)			
A – Complete set	2.34 $\pm$ 0.38	6.90 $\pm$ 1.22	469.31 $\pm$ 51.79	462.51 $\pm$ 43.06
B – Complete set	1.21 $\pm$ 0.33	3.31 $\pm$ 0.96	94.85 $\pm$ 16.42	93.66 $\pm$ 11.49
C – Complete set	2.76 $\pm$ 0.63	5.98 $\pm$ 2.12	887.63 $\pm$ 80.10	881.02 $\pm$ 86/34
A – Less reference	2.42 $\pm$ 0.49	5.23 $\pm$ 0.86	438.60 $\pm$ 35.07	430.05 $\pm$ 33.43
B – Less reference	1.12 $\pm$ 0.23	2.61 $\pm$ 0.50	82.50 $\pm$ 9.90	82.46 $\pm$ 8.79
C – Less reference	2.77 $\pm$ 0.69	5.21 $\pm$ 1.21	803.73 $\pm$ 68.61	789.93 $\pm$ 59.35
A – Less anchor	2.53 $\pm$ 0.82	7.75 $\pm$ 2.47	345.49 $\pm$ 75.84	322.77 $\pm$ 90.00
B – Less anchor	1.17 $\pm$ 0.36	3.29 $\pm$ 1.33	55.30 $\pm$ 17.19	52.69 $\pm$ 13.13
C – Less anchor	4.30 $\pm$ 0.69	9.88 $\pm$ 1.19	644.31 $\pm$ 65.67	630.38 $\pm$ 67.62
A – Less training	4.00 $\pm$ 0.40	12.98 $\pm$ 0.86	131.44 $\pm$ 15.88	136.53 $\pm$ 13.16
B – Less training	2.01 $\pm$ 0.60	5.50 $\pm$ 1.51	68.33 $\pm$ 7.63	69.80 $\pm$ 6.36
C – Less training	4.17 $\pm$ 0.61	10.30 $\pm$ 0.78	215.96 $\pm$ 30.83	234.04 $\pm$ 15.66

3.5 Summary

This chapter presented a new approach in location fingerprinting that combines the methodology of LANDMARC and LEMT. The proposed method, CO-LEMT, uses functional relationship modeling as in LEMT, but with continuous output as in LANDMARC. The continuous-output characteristic of CO-LEMT allows it to achieve good accuracy when training points are not as dense as testing points. This scenario is more practical in real life because manual collection of training points is a tedious task.

The advantage of CO-LEMT over LANDMARC and LEMT can be clearly observed when less training points are used. The error of CO-LEMT does not increase as high as LEMT when less training points are used. On the other hand, LANDMARC has a much higher error when less reference devices are used. The proposed CO-LEMT consistently produces the lowest error across all datasets and all scenarios when compared with other location fingerprinting methods. In most datasets, we observed 0.5 m to 2 m improvement in error percentile from the existing location fingerprinting methods.

We observed a strong indication of overfitting on LEMT for dataset B, where scenarios with zero error under spatial variation suddenly have an error increase under spatial-temporal variation. This is most possibly caused by the record-and-compare mechanism used in LEMT. Such increase also happens to our proposed CO-LEMT, but not as large as in LEMT.

Chapter 4

Extreme Learning Machine for Location Fingerprinting

4.1 Introduction

In Chapter 3, we have introduced a new method in location fingerprinting named CO-LEMT that incorporates both M5 model tree and LANDMARC to achieve better accuracy with less number of devices and less calibration effort. However, processing time remains a critical issue carried forward from LEMT because CO-LEMT still uses M5 model tree that requires long training time and testing time.

Among non-parametric models, Extreme Learning Machine (ELM) has been known to have good generalization and quick processing time in modeling nonlinear regression and classification problems.

There have been some efforts in applying ELM as an indoor positioning system. However, they seem to be sporadic and lacking in comprehensive study on issues pertaining to the topology or configuration of ELM. Letchner, Fox, and LaMarca [87] introduced clustering of RSS fingerprints based on their similarities before feeding them into an ELM regression model, where the fingerprints of the tracking tag collected during training phase and the corresponding coordinates

are used as training data. This model may perform well in static environment, but they do not anticipate the possibilities of having RSS fingerprints that differ from those collected during training phase. Zou et al. proposed a combination of Weighted Path Loss (WPL) and ELM regression [88], where WPL helps to coarsely determine the zone in which the tracking tag is located to reduce the training time. The RSS data from reference tags during training phase and the corresponding coordinates are then used to train the ELM.

Although ELM is a black-box neural network approach which may seem to be a viable solution, we believe that there are many possibilities on how we can configure the ELM as location fingerprinting system. There are many issues pertaining to the training data for the ELM that warrant consideration. For example, training input of ELM may come from the reference tags only, from the tracking tag only, or from all reference tags and tracking tag. Since RSS data may also be collected during training phase and testing phase, it is also important to choose which phase should provide data for the training process. Furthermore, there are a variety of choices of ELM target output and configurations. How the choice of the model selected affects the accuracy and the processing speed of the indoor positioning system has not been explored.

In this chapter, we first present an overview of ELM and our new framework in applying ELM for location fingerprinting. Next, we present our approach in applying ELM on LEMT and CO-LEMT as an effort to reduce the processing time.

4.2 Extreme Learning Machine

Huang, Zhu, and Siew introduced Extreme Learning Machine (ELM) [54], a single hidden-layer feedforward network that do away with conventional iterative learning in order to have quick training time and good generalization. The ELM structure, as illustrated in Figure 4.1, consists of a layer of input neurons, a layer of hidden neurons, and a layer of output neurons.

Figure 4.1: Extreme Learning Machine (ELM) architecture

ELM uses supervised learning framework in which input data and their target output are provided during training phase. As opposed to conventional neural network learning methods, the weights of ELM hidden neurons are simply randomized and not optimized during training.

The output layer is the only layer optimized in the training phase. The weights of the output neurons β , are solved through least-square optimization as defined by:

$$\beta = H^+T \quad (5.1)$$

where $\mathbf{H}$ is the output vector of the hidden layer, $+$ is Moore-Penrose pseudo-inverse operator and $\mathbf{T}$ is the target output of the training data.

The pseudo-inverse operation needs to be computed differently when the problem is under-determined, i.e. when the number of training samples, N_d , is less than the number of hidden neurons, N_h . The computation of pseudo-inverse operation under different cases are defined as:

$$\beta = \begin{cases} (H^T H)^{-1} H^T T & \text{if } N_d \geq N_h \\ H^T (H H^T)^{-1} T & \text{if } N_d < N_h \end{cases} \quad (5.2)$$

If the $\mathbf{H}$ matrix is ill-conditioned, the computation of β could face a stability issue. Huang et al. improved the stability of ELM by applying Tikhonov regularization [89], where a small regularization constant is added along the diagonal elements of the matrix before inverting the matrix:

$$\beta = \begin{cases} \left(\frac{I}{C} + H^T H \right)^{-1} H^T T & \text{if } N_d \geq N_h \\ H^T \left(\frac{I}{C} + H H^T \right)^{-1} T & \text{if } N_d < N_h \end{cases} \quad (5.3)$$

The universal approximation capability of ELM has been proven through a comprehensive theoretical analysis presented by Huang, Chen, and Siew [90]. Also, experiments involving various classification and regression problems prove that the absence of hidden neuron optimization does not degrade the accuracy of ELM in comparison with conventional neural networks, despite cutting down the training time significantly [89].

4.3 Location fingerprinting translation framework

To systematically translate location fingerprinting methods into ELM, we need to list the information to use for each fingerprinting method and formulate the equivalent ELM configuration.

We formulate a new framework on how to translate a location fingerprinting technique into ELM using the following steps:

1. List all possible types of information that are related to the location fingerprinting problem.
2. Identify the types of information that are used anywhere in the location fingerprinting technique.
3. Understand the role of each type of information in the location fingerprinting technique.
4. Categorize the roles of the information and analyze whether they are suitable for training input, training target, testing input, or testing output.
5. Consider the properties of training target and testing output to decide the ELM type (i.e. as a regression or classification module). If the training target and testing output are in continuous space, ELM must be configured as a regression module. If they are in discrete space, ELM can either be configured as a regression or classification module, although classification is a more intuitive choice.

By referring to the mapping of each type of information presented in Table 3.1, the ELM configuration can be derived for each location fingerprinting method. Table 4.1 presents the selection of ELM type, either as regression or classification module, and the list of information to use during training phase and testing phase.

Table 4.1: Configuration of ELM-based location fingerprinting methods
Note: For NN method, there are 2 possible options:
regression (marked *) or classification (marked **)

Types of information	Fingerprinting method		
	NN (Regression* or classification**)	LANDMARC (Regression)	LEMT (Regression)
Training phase			
Reference RSS	-	-	Training input
Reference location coordinate	-	-	-
Tracking RSS	Training input	-	Training target
Tracking location coordinate	Training target*	-	-
Tracking location index	Training target**		Model indexing
Testing phase			
Reference RSS	-	Training input	Testing input
Reference location coordinate	-	Training target	-
Tracking RSS	Testing input	Training input	Testing output
Tracking location coordinate	Testing output*	Testing output	-
Tracking location index	Testing output**	-	-

In NN method, the RSS fingerprints of tracking device obtained during training phase are used as the training input of the system, whereas their corresponding location indices are used as the training target. Then, during testing phase, as the tracking device records the current RSS fingerprint, we compare the fingerprint with the recorded training input, find the best match, and retrieve the coordinate of the best-match location index as the location estimate. In machine learning paradigm, this compare-and-select method can be implemented as supervised classification, where the fingerprints are classified based on their location indices. Another possible equivalent machine learning method of NN is supervised regression, where RSS fingerprints are the training input and their corresponding location coordinates are the training target. The main difference of those two methods is that the classification produces discrete output space whereas the regression produces continuous output space.

In LANDMARC, fingerprints during training phase are not recorded since it is not used in the method. If we were to use machine learning to represent LANDMARC, the machine learning needs to be trained with some fingerprints collected during testing phase, which come from the reference devices. The RSS of reference devices and their location coordinates are to be used as training inputs and training targets, respectively. Then, upon receiving the RSS of the tracking device, the machine learning would estimate the location coordinates of the tracking device. Since the location of the reference device are likely to be different from the location of the tracking device, we must set the machine

learning mode as regression to allow it to interpolate or extrapolate the location coordinates from the coordinates of the reference devices.

In LEMT, multiple functional relationship models are required to represent each training location. Therefore, we need to use the location index of the tracking device during training phase for model indexing purpose. Since each model in LEMT estimates the RSS of the tracking device from the RSS of the reference devices, the equivalent machine learning representation of the LEMT would be a regression model that accepts the RSS of reference devices and the RSS of tracking devices collected during training phase as the training inputs and training targets, respectively.

The proposed framework makes it easier to translate any location fingerprinting system into an ELM-based technique. It may also help to analyze what information has not been included in a location fingerprinting technique and how the fingerprinting technique can be developed further.

4.4 Methodology of ELM-based LEMT

As discussed in Chapter 3, LEMT and CO-LEMT require much longer processing time than NN and LANDMARC. This happens because the methods rely on M5 model tree as the functional relationship model. Here we propose a new location fingerprinting method that uses ELM as the functional relationship

model in LEMT and CO-LEMT. The ELM-based method derived from LEMT is named as Location Estimation using ELM (LEELM), whereas its continuous-output form is named as Continuous-output LEELM (CO-LEELM).

Referring to Table 4.1, converting LEMT into ELM-based technique requires ELM to be configured as regression model. One concern with this configuration is that missing data may exist in the fingerprints and affect the performance of the ELM regression. In M5 model tree which is formed by multiple piecewise-linear regression functions, the missing data would be automatically excluded from the model tree. In ELM, the final representation of the model consists of one nonlinear regression function that represents the functional mapping between the input and output, and thus the presence of missing data could affect the accuracy of the model. It is necessary to evaluate whether the handling of the missing data, either through imputation or filtering, without compromising the accuracy of the ELM. To address this concern, we constructed two additional scenarios where missing data are filtered out rather than imputed. The scenarios are: 1) Filtered LEELM (F-LEELM), which is derived from LEELM, and 2) Filtered Continuous-output LEELM (FCO-LEELM).

The naming convention we use for ELM-based proposed methods is presented in Table 4.2. For imputation, we used a low RSS value to replace the missing RSS in the fingerprints. We chose a low RSS as a replacement value because the absence of an RSS value is likely to be caused by signal loss due to distance to anchor that is too far away.

Table 4.2: Description of proposed ELM-based location fingerprinting methods

No	Method name	Non-ELM equivalent method	Treatment for missing data
1.	LEELM	LEMT	Imputation
2.	CO-LEELM	CO-LEMT	Imputation
3.	F-LEELM	LEMT	Removal
4.	FCO-LEELM	CO-LEMT	Removal

4.5 Performance evaluation

4.5.1 Experimental setup

To evaluate the performance of our proposed methods, we use the same datasets and scenarios presented in Section 3.4, and apply them on our proposed methods. CO-LEMT is used as a benchmark method in the performance evaluation of this chapter.

As suggested by Huang, Zhu, and Siew [54], we normalized the range of ELM input and output to range from 0 to 1. The RSS of RFID devices are represented as an integer value ranging from 0 to 255 of Link Quality Indicator (LQI) as stated in IEEE 802.15.4 standard for RFID [91]. The mapping of the LQI into dBm is decided by the RFID manufacturer, which in our case is unknown since we could not find it stated in the device’s user manual. Nevertheless, we can still normalize the values without knowing its mapping to dBm. Since the range of the LQI is from 0 to 255, we normalize the values by dividing them by 255.

4.5.2 Selection of parameters

To determine the ELM parameters, we used sigmoid activation function and varied the number of hidden neurons and the regularization constant. The parameters were chosen based on the optimum performance of the datasets under spatial temporal variation when using a complete set. The datasets are used to find the optimal number of hidden neurons and regularization parameter of ELM. The number of hidden neurons, N_h , is chosen from $\{5, 10, 50, 100, 500\}$, whereas the regularization parameter, C , is chosen from $\{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$.

When choosing the number of hidden neurons, we searched for the smallest number of hidden neurons that can still achieve good accuracy. Higher number of hidden neurons is not preferred because it results in higher complexity and longer training time and testing time. As for the regularization parameter, we selected the one that produces the best accuracy. Training time and testing time are not considered when choosing the regularization parameter because the choice of regularization parameter does not affect system complexity.

As shown in Figure 4.2 and Figure 4.3, training time and testing time of CO-LEELM appear to be relatively constant for $N_h = 50$ and below. Longer training time and testing time are observable when $N_h = 100$, followed by a significant increase when $N_h = 500$.

Figure 4.2: Training time of CO-LEELM under different numbers of hidden neurons, N_h , and different regularization parameter, C

Figure 4.3: Testing time of CO-LEELM under different numbers of hidden neurons, N_h , and different regularization parameter, C

In terms of accuracy, Figure 4.4, Figure 4.5, and Figure 4.6 show that the number of hidden neurons and the value of regularization parameter affect the accuracy significantly. Based on the results of the three datasets, the lowest error is obtained when C value is 10^0 , 10^1 , and 10^2 , regardless the value of N_h .

Figure 4.4: Accuracy of CO-LEELM for Dataset A under different numbers of hidden neurons, N_h , and different regularization parameter, C

Figure 4.5: Accuracy of CO-LEELM for Dataset B under different numbers of hidden neurons, N_h , and different regularization parameter, C

Figure 4.6: Accuracy of CO-LEELM for Dataset C under different numbers of hidden neurons, N_h , and different regularization parameter, C

Considering the processing time and accuracy, the ELM parameters used in the following experiments are as described in Table 4.3. The choice of $N_h = 50$ results in relatively short training time and testing time, whereas $C = 10^1$ is one of the C values that produce the lowest error.

Table 4.3: Final choice of ELM parameters

Algorithm	Activation function	Number of hidden neurons	Regularization parameter	Type
ELM	Sigmoid	50	10^1	Regression

4.5.3 Performance result under spatial variation

Table 4.4 shows the 90th and 95th error percentile under spatial variation.

Table 4.4: Accuracy of ELM-based methods under spatial variation

Dataset– Scenario	Method				
	CO-LEMT	LEELM	CO-LEELM	F-LEELM	FCO-LEELM
	90th error percentile in meters (mean $\pm$ standard deviation)				
A – Complete set	1.29 $\pm$ 0.05	2.15 $\pm$ 0.21	1.70 $\pm$ 0.21	2.35 $\pm$ 0.39	1.79 $\pm$ 0.30
B – Complete set	0.26 $\pm$ 0.31	0.64 $\pm$ 0.93	0.92 $\pm$ 0.88	1.20 $\pm$ 1.73	1.30 $\pm$ 1.42
C – Complete set	1.25 $\pm$ 0.09	1.92 $\pm$ 0.13	1.51 $\pm$ 0.10	2.02 $\pm$ 0.03	1.61 $\pm$ 0.10
A – Less reference	1.57 $\pm$ 0.10	2.39 $\pm$ 0.41	1.77 $\pm$ 0.25	2.38 $\pm$ 0.38	1.81 $\pm$ 0.33
B – Less reference	0.65 $\pm$ 0.81	1.11 $\pm$ 1.61	1.09 $\pm$ 1.13	1.20 $\pm$ 1.73	1.22 $\pm$ 1.30
C – Less reference	1.30 $\pm$ 0.06	1.97 $\pm$ 0.07	1.49 $\pm$ 0.12	2.03 $\pm$ 0.02	1.64 $\pm$ 0.15
A – Less anchor	1.54 $\pm$ 0.13	2.57 $\pm$ 0.46	2.03 $\pm$ 0.33	2.69 $\pm$ 0.39	2.21 $\pm$ 0.31
B – Less anchor	0.18 $\pm$ 0.12	0.85 $\pm$ 1.16	1.12 $\pm$ 0.85	1.70 $\pm$ 2.26	1.36 $\pm$ 1.22
C – Less anchor	1.69 $\pm$ 0.08	2.51 $\pm$ 0.33	1.98 $\pm$ 0.21	2.58 $\pm$ 0.28	2.00 $\pm$ 0.20
A – Less training	2.84 $\pm$ 0.26	3.93 $\pm$ 0.39	2.71 $\pm$ 0.25	3.90 $\pm$ 0.41	2.73 $\pm$ 0.22
B – Less training	4.03 $\pm$ 0.36	5.01 $\pm$ 1.33	4.26 $\pm$ 0.13	5.01 $\pm$ 1.33	4.27 $\pm$ 0.13
C – Less training	2.63 $\pm$ 0.13	3.55 $\pm$ 0.19	2.48 $\pm$ 0.07	3.57 $\pm$ 0.05	2.49 $\pm$ 0.11
	95th error percentile in meters (mean $\pm$ standard deviation)				
A – Complete set	1.69 $\pm$ 0.10	2.81 $\pm$ 0.30	2.12 $\pm$ 0.35	3.00 $\pm$ 0.34	2.34 $\pm$ 0.32
B – Complete set	0.50 $\pm$ 0.51	1.25 $\pm$ 1.81	1.62 $\pm$ 1.37	2.20 $\pm$ 3.16	1.83 $\pm$ 1.64
C – Complete set	1.67 $\pm$ 0.08	2.58 $\pm$ 0.35	2.05 $\pm$ 0.24	2.91 $\pm$ 0.19	2.20 $\pm$ 0.18
A – Less reference	1.91 $\pm$ 0.15	2.90 $\pm$ 0.41	2.27 $\pm$ 0.34	3.01 $\pm$ 0.33	2.36 $\pm$ 0.34
B – Less reference	1.30 $\pm$ 1.68	1.86 $\pm$ 2.67	1.80 $\pm$ 1.59	2.20 $\pm$ 3.16	1.86 $\pm$ 1.67
C – Less reference	1.63 $\pm$ 0.10	2.80 $\pm$ 0.23	2.03 $\pm$ 0.20	2.91 $\pm$ 0.19	2.18 $\pm$ 0.21
A – Less anchor	1.98 $\pm$ 0.04	3.38 $\pm$ 0.49	2.54 $\pm$ 0.22	3.56 $\pm$ 0.62	2.67 $\pm$ 0.20
B – Less anchor	0.58 $\pm$ 0.43	2.24 $\pm$ 1.98	1.68 $\pm$ 1.41	3.47 $\pm$ 3.67	2.10 $\pm$ 2.05
C – Less anchor	2.06 $\pm$ 0.04	3.20 $\pm$ 0.34	2.48 $\pm$ 0.37	3.34 $\pm$ 0.56	2.55 $\pm$ 0.22
A – Less training	3.33 $\pm$ 0.34	4.78 $\pm$ 0.25	3.21 $\pm$ 0.13	4.82 $\pm$ 0.37	3.19 $\pm$ 0.09
B – Less training	5.08 $\pm$ 0.69	6.08 $\pm$ 1.69	4.87 $\pm$ 0.78	6.08 $\pm$ 1.69	4.87 $\pm$ 0.79
C – Less training	3.03 $\pm$ 0.13	4.54 $\pm$ 1.01	2.81 $\pm$ 0.14	4.76 $\pm$ 0.86	2.95 $\pm$ 0.21

The error percentile statistics show that the accuracy of LEELM and F-LEELM experiences significant degradation from the accuracy of CO-LEMT. However, there are some scenarios where the errors of LEELM are extremely low, which most likely indicate overfitting. Applying continuous-output, as in CO-LEELM and FCO-LEELM, improves the accuracy in most scenarios, although they are still slightly worse than CO-LEMT. The error percentile differences between CO-LEMT and the continuous-output ELM-based methods, CO-LEELM and FCO-LEELM, are mostly about 0.1 m to 0.5 m.

(a) Dataset A – Complete set

(b) Dataset A – Less reference

(c) Dataset A – Less anchor

(d) Dataset A – Less training

Figure 4.7: Cumulative error distribution of ELM-based methods for Dataset A under spatial variation

The choice of using imputation or filtering method on the continuous-output ELM techniques does not appear to affect the accuracy quite much. This can be observed in Figure 4.7, Figure 4.8, and Figure 4.9 where the errors of CO-LEELM and FCO-LEELM intersect with each other most of the time. Although the cumulative distribution plots of CO-LEELM sometimes appear to be better than FCO-LEELM, the error statistics in Table 4.4 do not seem to indicate a significant difference between the two methods.

Figure 4.8: Cumulative error distribution of ELM-based methods for Dataset B under spatial variation

(a) Dataset C – Complete set

(b) Dataset C – Less reference

(c) Dataset C – Less anchor

(d) Dataset C – Less training

Figure 4.9: Cumulative error distribution of ELM-based methods for Dataset C under spatial variation

In terms of processing time, as shown in Table 4.5, there is a significant speed improvement of ELM-based methods as compared to CO-LEMT. The largest speed improvement is experienced by CO-LEELM. Its training time is about 15 to 30 times faster than CO-LEMT, whereas its testing time is about 6 to 10 times faster.

From the results we obtained under spatial variation, considering that the accuracy of the continuous-output ELM methods is only slightly lower than CO-LEMT, there is a strong indication that ELM can be used to replace M5 model

tree as functional relationship model in the location fingerprinting method with much faster speed while maintaining satisfying accuracy. This observation will be verified with the results under spatial-temporal variation we will present next.

Table 4.5: Processing time of ELM-based methods under spatial variation

Dataset–Scenario	Method				
	CO-LEMT	LEELM	CO-LEELM	F-LEELM	FCO-LEELM
	Training time in milliseconds (mean $\pm$ standard deviation)				
A – Complete set	921.86 $\pm$ 214.48	32.15 $\pm$ 8.20	29.06 $\pm$ 7.27	48.34 $\pm$ 12.31	44.41 $\pm$ 9.31
B – Complete set	324.54 $\pm$ 42.89	10.96 $\pm$ 1.62	10.98 $\pm$ 2.48	15.88 $\pm$ 1.55	16.12 $\pm$ 2.50
C – Complete set	1078.43 $\pm$ 106.49	52.95 $\pm$ 2.86	52.08 $\pm$ 1.35	76.70 $\pm$ 19.41	70.21 $\pm$ 2.22
A – Less reference	456.75 $\pm$ 34.64	24.78 $\pm$ 3.48	24.25 $\pm$ 1.85	32.44 $\pm$ 5.41	31.40 $\pm$ 1.60
B – Less reference	226.74 $\pm$ 35.23	10.56 $\pm$ 0.88	10.36 $\pm$ 0.87	13.64 $\pm$ 1.86	13.85 $\pm$ 2.52
C – Less reference	660.31 $\pm$ 79.43	47.16 $\pm$ 1.52	46.76 $\pm$ 1.56	58.04 $\pm$ 2.56	58.16 $\pm$ 2.00
A – Less anchor	434.60 $\pm$ 26.41	15.06 $\pm$ 0.69	14.83 $\pm$ 0.83	22.83 $\pm$ 1.82	22.55 $\pm$ 1.36
B – Less anchor	159.10 $\pm$ 20.17	5.84 $\pm$ 0.60	5.69 $\pm$ 0.78	7.75 $\pm$ 0.57	7.79 $\pm$ 1.08
C – Less anchor	399.28 $\pm$ 35.10	19.72 $\pm$ 1.42	19.35 $\pm$ 1.31	27.49 $\pm$ 4.52	26.35 $\pm$ 2.59
A – Less training	104.83 $\pm$ 13.46	4.01 $\pm$ 0.27	3.65 $\pm$ 0.29	5.95 $\pm$ 0.50	5.70 $\pm$ 0.51
B – Less training	124.25 $\pm$ 5.99	4.26 $\pm$ 0.44	4.12 $\pm$ 0.40	5.86 $\pm$ 0.58	5.72 $\pm$ 1.59
C – Less training	120.75 $\pm$ 19.86	7.30 $\pm$ 0.34	6.96 $\pm$ 0.49	9.68 $\pm$ 0.60	9.49 $\pm$ 0.59
	Testing time in milliseconds (mean $\pm$ standard deviation)				
A – Complete set	595.38 $\pm$ 102.91	58.34 $\pm$ 13.43	52.97 $\pm$ 9.48	121.80 $\pm$ 24.22	118.33 $\pm$ 27.49
B – Complete set	102.46 $\pm$ 20.38	12.91 $\pm$ 1.49	13.14 $\pm$ 1.28	27.16 $\pm$ 2.14	28.53 $\pm$ 3.09
C – Complete set	1056.52 $\pm$ 82.73	86.53 $\pm$ 1.78	83.39 $\pm$ 2.75	209.85 $\pm$ 25.64	203.39 $\pm$ 9.10
A – Less reference	464.46 $\pm$ 64.56	47.15 $\pm$ 3.28	43.36 $\pm$ 2.33	101.98 $\pm$ 5.41	100.04 $\pm$ 7.67
B – Less reference	93.19 $\pm$ 9.34	12.09 $\pm$ 1.18	12.57 $\pm$ 1.03	26.49 $\pm$ 5.77	26.31 $\pm$ 2.35
C – Less reference	796.18 $\pm$ 44.54	76.18 $\pm$ 2.37	74.20 $\pm$ 2.56	184.27 $\pm$ 11.52	180.81 $\pm$ 8.92
A – Less anchor	280.48 $\pm$ 23.97	35.07 $\pm$ 1.69	31.58 $\pm$ 1.62	69.55 $\pm$ 4.08	65.91 $\pm$ 3.52
B – Less anchor	54.60 $\pm$ 6.98	8.72 $\pm$ 0.99	9.00 $\pm$ 1.27	15.85 $\pm$ 1.48	16.41 $\pm$ 2.62
C – Less anchor	354.33 $\pm$ 12.95	40.24 $\pm$ 2.83	36.37 $\pm$ 1.15	88.96 $\pm$ 20.79	84.20 $\pm$ 8.18
A – Less training	72.67 $\pm$ 3.28	7.59 $\pm$ 0.57	10.74 $\pm$ 0.57	15.94 $\pm$ 1.27	19.53 $\pm$ 1.75
B – Less training	40.72 $\pm$ 3.54	4.97 $\pm$ 0.52	6.32 $\pm$ 0.59	10.37 $\pm$ 1.04	12.33 $\pm$ 2.78
C – Less training	123.88 $\pm$ 11.37	11.92 $\pm$ 0.38	15.13 $\pm$ 0.88	28.98 $\pm$ 2.07	32.53 $\pm$ 3.28

4.5.4 Performance result under spatial-temporal variation

The results under spatial-temporal variation are presented in Table 4.6.

Table 4.6: Accuracy of ELM-based methods under spatial-temporal variation

Dataset–Scenario	Method				
	CO-LEMT	LEELM	CO-LEELM	F-LEELM	FCO-LEELM
	90th error percentile in meters (mean $\pm$ standard deviation)				
A – Complete set	2.36 $\pm$ 0.16	2.93 $\pm$ 0.19	2.22 $\pm$ 0.18	2.85 $\pm$ 0.23	2.22 $\pm$ 0.16
B – Complete set	3.67 $\pm$ 0.53	2.36 $\pm$ 2.06	3.24 $\pm$ 0.62	1.20 $\pm$ 1.17	3.30 $\pm$ 0.65
C – Complete set	2.10 $\pm$ 0.23	2.48 $\pm$ 0.28	1.91 $\pm$ 0.19	2.54 $\pm$ 0.35	1.91 $\pm$ 0.16
A – Less reference	2.38 $\pm$ 0.23	2.88 $\pm$ 0.25	2.25 $\pm$ 0.14	2.81 $\pm$ 0.26	2.30 $\pm$ 0.16
B – Less reference	3.67 $\pm$ 0.54	2.59 $\pm$ 2.12	3.27 $\pm$ 0.64	2.62 $\pm$ 2.43	3.49 $\pm$ 0.74
C – Less reference	2.03 $\pm$ 0.24	2.47 $\pm$ 0.28	1.90 $\pm$ 0.20	2.53 $\pm$ 0.36	1.89 $\pm$ 0.16
A – Less anchor	2.60 $\pm$ 0.23	3.13 $\pm$ 0.31	2.50 $\pm$ 0.17	3.18 $\pm$ 0.27	2.52 $\pm$ 0.21
B – Less anchor	3.36 $\pm$ 0.69	2.72 $\pm$ 1.75	3.30 $\pm$ 0.70	3.36 $\pm$ 1.24	3.37 $\pm$ 0.78
C – Less anchor	2.46 $\pm$ 0.17	2.94 $\pm$ 0.19	2.27 $\pm$ 0.13	2.87 $\pm$ 0.21	2.25 $\pm$ 0.14
A – Less training	2.95 $\pm$ 0.39	3.85 $\pm$ 0.35	2.67 $\pm$ 0.12	3.73 $\pm$ 0.37	2.62 $\pm$ 0.16
B – Less training	4.10 $\pm$ 0.24	4.18 $\pm$ 1.63	4.26 $\pm$ 0.21	3.96 $\pm$ 1.35	4.21 $\pm$ 0.12
C – Less training	2.85 $\pm$ 0.60	3.87 $\pm$ 0.53	2.56 $\pm$ 0.16	3.74 $\pm$ 0.31	2.66 $\pm$ 0.15
	95th error percentile in meters (mean $\pm$ standard deviation)				
A – Complete set	2.86 $\pm$ 0.20	3.40 $\pm$ 0.40	2.74 $\pm$ 0.19	3.38 $\pm$ 0.50	2.79 $\pm$ 0.30
B – Complete set	4.54 $\pm$ 0.62	5.05 $\pm$ 2.90	4.29 $\pm$ 1.05	4.54 $\pm$ 2.91	4.22 $\pm$ 1.14
C – Complete set	2.81 $\pm$ 0.45	3.12 $\pm$ 0.32	2.54 $\pm$ 0.30	3.34 $\pm$ 0.63	2.57 $\pm$ 0.42
A – Less reference	2.91 $\pm$ 0.14	3.28 $\pm$ 0.28	2.74 $\pm$ 0.17	3.25 $\pm$ 0.36	2.78 $\pm$ 0.29
B – Less reference	4.25 $\pm$ 0.48	5.42 $\pm$ 2.63	4.31 $\pm$ 1.12	4.67 $\pm$ 2.79	4.23 $\pm$ 1.14
C – Less reference	2.69 $\pm$ 0.38	3.14 $\pm$ 0.35	2.51 $\pm$ 0.26	3.30 $\pm$ 0.59	2.55 $\pm$ 0.40
A – Less anchor	3.01 $\pm$ 0.22	3.76 $\pm$ 0.37	2.89 $\pm$ 0.23	3.74 $\pm$ 0.45	2.91 $\pm$ 0.29
B – Less anchor	3.66 $\pm$ 0.92	4.51 $\pm$ 2.12	3.68 $\pm$ 0.57	3.85 $\pm$ 1.30	3.51 $\pm$ 0.75
C – Less anchor	2.96 $\pm$ 0.17	3.43 $\pm$ 0.37	2.94 $\pm$ 0.34	3.41 $\pm$ 0.44	2.94 $\pm$ 0.31
A – Less training	3.38 $\pm$ 0.41	5.12 $\pm$ 0.70	3.05 $\pm$ 0.25	5.09 $\pm$ 0.68	2.97 $\pm$ 0.23
B – Less training	4.51 $\pm$ 0.42	5.13 $\pm$ 2.63	4.61 $\pm$ 0.41	5.16 $\pm$ 2.61	4.61 $\pm$ 0.42
C – Less training	3.27 $\pm$ 0.71	5.25 $\pm$ 0.75	3.15 $\pm$ 0.35	5.26 $\pm$ 0.77	3.09 $\pm$ 0.31

Among the proposed methods, CO-LEELM and FCO-LEELM offer the best accuracy. The 90th and 95th error percentile of CO-LEELM and FCO-LEELM are only 0.1 m to 0.5 m, at most, higher than CO-LEMT.

Figure 4.10: Cumulative error distribution of ELM-based methods for Dataset A under spatial-temporal variation

As shown in Figure 4.10, Figure 4.11, and Figure 4.12, the errors of those methods are comparable with CO-LEMT in most scenarios, except for dataset A and B with less training data where the ELM-based methods perform better than CO-LEMT.

The results indicate that ELM could be used as a functional relationship models in place of the M5 model tree originally used in LEMT.

Figure 4.11: Cumulative error distribution of ELM-based methods for Dataset B under spatial-temporal variation

(a) Dataset C – Complete set

(b) Dataset C – Less reference

(c) Dataset C – Less anchor

(d) Dataset C – Less training

Figure 4.12: Cumulative error distribution of ELM-based methods for Dataset C under spatial-temporal variation

In terms of training time and testing time, as shown in Table 4.7, the results under spatial-temporal variation are generally similar with the ones obtained under spatial variation. LEELM and CO-LEELM have the shortest time, followed by the filtering-based methods, F-LEELM and FCO-LEELM. The speed improvement of LEELM and CO-LEELM is about 20 to 40 times for training time, and about 6 to 10 times for testing time.

Table 4.7: Training time and testing time of ELM-based methods
under spatial-temporal variation

Dataset–Scenario	Method				
	CO-LEMT	LEELM	CO-LEELM	F-LEELM	FCO-LEELM
	Training time in milliseconds (mean $\pm$ standard deviation)				
A – Complete set	995.46 $\pm$ 122.64	26.87 $\pm$ 5.81	26.40 $\pm$ 5.07	41.65 $\pm$ 10.18	40.81 $\pm$ 6.78
B – Complete set	322.71 $\pm$ 44.51	11.68 $\pm$ 3.18	11.95 $\pm$ 3.30	16.28 $\pm$ 2.64	15.99 $\pm$ 2.90
C – Complete set	1282.28 $\pm$ 183.58	53.72 $\pm$ 7.28	54.03 $\pm$ 7.53	74.19 $\pm$ 12.66	73.15 $\pm$ 10.98
A – Less reference	541.70 $\pm$ 57.23	26.61 $\pm$ 3.50	26.06 $\pm$ 3.27	34.49 $\pm$ 4.26	34.12 $\pm$ 4.24
B – Less reference	210.86 $\pm$ 29.13	10.94 $\pm$ 1.71	10.90 $\pm$ 1.90	13.77 $\pm$ 2.07	13.71 $\pm$ 2.41
C – Less reference	844.00 $\pm$ 109.85	53.36 $\pm$ 7.47	54.75 $\pm$ 11.42	65.55 $\pm$ 8.99	65.21 $\pm$ 8.34
A – Less anchor	673.60 $\pm$ 208.97	18.82 $\pm$ 6.99	18.27 $\pm$ 5.05	29.04 $\pm$ 10.00	28.75 $\pm$ 12.39
B – Less anchor	174.58 $\pm$ 38.11	6.70 $\pm$ 3.19	6.17 $\pm$ 2.23	8.97 $\pm$ 2.76	9.49 $\pm$ 5.30
C – Less anchor	888.21 $\pm$ 100.69	40.43 $\pm$ 22.99	39.22 $\pm$ 13.21	51.62 $\pm$ 6.73	50.73 $\pm$ 5.61
A – Less training	245.39 $\pm$ 31.04	7.90 $\pm$ 0.89	7.59 $\pm$ 0.76	11.75 $\pm$ 1.00	11.38 $\pm$ 0.92
B – Less training	235.23 $\pm$ 19.37	8.06 $\pm$ 2.40	7.77 $\pm$ 1.59	10.96 $\pm$ 0.76	10.78 $\pm$ 1.01
C – Less training	298.52 $\pm$ 40.71	14.72 $\pm$ 1.01	14.31 $\pm$ 1.05	19.54 $\pm$ 1.30	19.24 $\pm$ 1.42
	Testing time in milliseconds (mean $\pm$ standard deviation)				
A – Complete set	462.51 $\pm$ 43.06	52.59 $\pm$ 9.20	47.68 $\pm$ 5.34	112.46 $\pm$ 14.42	110.46 $\pm$ 18.36
B – Complete set	93.66 $\pm$ 11.49	13.96 $\pm$ 3.57	14.42 $\pm$ 3.10	28.94 $\pm$ 5.51	28.38 $\pm$ 3.87
C – Complete set	881.02 $\pm$ 86/34	87.85 $\pm$ 11.47	83.91 $\pm$ 11.35	216.73 $\pm$ 30.59	213.21 $\pm$ 28.86
A – Less reference	430.05 $\pm$ 33.43	50.50 $\pm$ 5.61	45.68 $\pm$ 4.36	111.04 $\pm$ 12.24	106.47 $\pm$ 10.69
B – Less reference	82.46 $\pm$ 8.79	12.79 $\pm$ 3.45	12.89 $\pm$ 2.34	26.05 $\pm$ 5.30	25.85 $\pm$ 3.21
C – Less reference	789.93 $\pm$ 59.35	86.31 $\pm$ 13.07	81.76 $\pm$ 10.51	206.66 $\pm$ 25.22	201.72 $\pm$ 23.86
A – Less anchor	322.77 $\pm$ 90.00	43.37 $\pm$ 15.09	36.99 $\pm$ 9.85	85.03 $\pm$ 24.15	80.64 $\pm$ 21.94
B – Less anchor	52.69 $\pm$ 13.13	9.97 $\pm$ 4.46	9.47 $\pm$ 2.32	17.18 $\pm$ 5.77	16.70 $\pm$ 3.80
C – Less anchor	630.38 $\pm$ 67.62	79.76 $\pm$ 8.89	71.19 $\pm$ 8.22	166.62 $\pm$ 19.65	160.53 $\pm$ 16.54
A – Less training	136.53 $\pm$ 13.16	15.12 $\pm$ 0.97	21.54 $\pm$ 1.42	31.87 $\pm$ 2.56	38.95 $\pm$ 3.14
B – Less training	69.80 $\pm$ 6.36	9.10 $\pm$ 1.68	12.39 $\pm$ 2.37	19.02 $\pm$ 1.35	22.03 $\pm$ 2.03
C – Less training	234.04 $\pm$ 15.66	23.83 $\pm$ 1.16	30.47 $\pm$ 1.80	57.41 $\pm$ 4.24	64.14 $\pm$ 4.89

4.6 Summary

In this chapter, we have presented a systematic framework on how ELM can be applied on the existing location fingerprinting techniques by observing the types of information used by the techniques. Also, we proposed a new approach in location fingerprinting where ELM regression is used as functional relationship model to achieve shorter training time and testing time.

The accuracy of the proposed method CO-LEELM and FCO-LEELM is comparable with CO-LEMT in most scenarios and sometimes better than CO-LEMT, particularly for scenarios that use less training points.

Incomplete data can be handled either through imputation or filtering. There is a slight increase in the training time and testing time of the F-LEELM and FCO-LEELM due to additional filtering process. However, we do not observe any significant accuracy difference between the imputation method and filtering method in handling incomplete data.

As for the output space selection, similar with the results obtained in Chapter 3, continuous output space produces better accuracy, especially for cases with less training points. A combination of imputation and continuous output, as applied on CO-LEELM, provides the best trade-off in terms of accuracy and processing time in the ELM-based equivalent algorithm for LEMT.

Chapter 5

Substitution of Reference Fingerprints

5.1 Introduction

In Chapter 4, we have further improved the continuous-output location fingerprinting we presented in Chapter 3 by using ELM as a functional relationship model instead of M5 model tree. When tested with our experimental data, the proposed CO-LEELM is found to have shorter processing speed and good adaptability to spatial-temporal variation.

Although some experimental data have been collected and used to assess the methods, it is essential for newly proposed methods to be tested using datasets collected by an independent third-party to confirm whether the proposed methods truly perform better than the existing methods under different system setup and environment conditions. However, finding such dataset to test our proposed methods is not a trivial task since our methods are derived from LEMT that requires not only offline fingerprints, but also reference devices, where the following requirements must be met:

1. RSS must be measured and recorded for several devices. At least one device is appointed as tracking device, whereas other devices serve as reference devices.

2. Reference devices must be placed at fixed locations at all time.
3. The location coordinates of the reference devices must be known.
4. RSS of reference devices must be continuously measured not only throughout offline phase, but also throughout online phase.

To the best of our knowledge, open-access datasets that fulfill those requirements are not available yet. In some datasets, RSS data are measured using only one device, which serves as a tracking device. Some datasets contain RSS data of several devices, but the devices are measured at non-fixed locations during offline and online phase, and thus the RSS cannot be treated as data from reference devices. As such, to facilitate the use of the open-access datasets, it is appropriate to alter our proposed methods such that they can operate in the absence of reference devices.

Other than allowing us to test the methods using open-access datasets, altering the proposed methods to work in system setup without reference devices have some practical benefits. Without reference devices, the device cost would automatically be reduced since we use less number of devices. If the tracking and reference devices are battery-operated, removing the reference devices means reducing the amount of maintenance work required to replace or recharge the batteries. This setup is also more acceptable for implementation in restricted venues, where placing reference devices at fixed locations might not be allowed due to rigorous security measures.

Since the presented location fingerprinting methods relies on RSS data from reference devices, those data must be replaced by another data that serve a similar purpose. This could be obtained by periodically collecting fingerprints at locations where reference devices are supposed to be placed. This approach, however, requires additional manpower to collect the additional data. Since additional manpower would increase cost, it is desirable to keep the number of additional data as little as possible.

In this chapter, we select an open-access dataset for indoor positioning that contains no reference devices to test the performance of our proposed methods. Since our proposed location fingerprinting methods requires reference fingerprints, we altered the data configuration where some fingerprints are used as reference fingerprints.

5.2 Open-access datasets

Although indoor positioning has been an emerging field for decades, publishing open-access datasets for RSS-based indoor positioning have only been prevalent in the last 4 years. Since the trend of publishing RSS-based datasets is still in its early stage, there is currently no formal procedure on how to setup the system and collect the data. Each dataset has distinct characteristics in many aspects, such as its system setup, the type of environment, the number of fingerprint samples, the length of data collection campaigns, and whether the coordinates of

data collection points repeat across different data collection campaigns. Unfortunately, the data collection methodology may affect the usability of the dataset since every location fingerprinting methods have unique needs.

In Table 5.1, we present some open-access datasets that collect RSS fingerprints within a prolonged time span, such as UJIIndoorLoc [92] [93], Crowdsourced Wi-Fi dataset [94] [95], BLE RSSI dataset [96] [97], and Long-term Wi-Fi dataset [98] [99]. Other than the ones listed in the table, we also found some other datasets [100] [101] which we exclude from the list because they were only collected during one day. Since our proposed method needs to be tested against spatial-temporal variation, it is necessary to choose an open-access dataset that is collected during a time span that is significantly longer than the datasets we used in the previous chapters to provide a more compelling performance evaluation. Among the shortlisted datasets in Table 5.1, Long-term Wi-Fi dataset [98] [99] is the longest dataset containing 15 months of fingerprint samples.

Besides, to substitute reference devices, we need the dataset to contain some fingerprints collected at fixed coordinates repeatedly over time, either collected using one device or multiple devices. This is necessary since in the location fingerprinting methods presented in Chapter 3 and Chapter 4, the reference devices are placed at fixed coordinates. Among the shortlisted datasets, only Long-term Wi-Fi dataset contains repeating coordinates. Therefore, we choose Long-term Wi-Fi dataset to evaluate the performance of our proposed method.

Table 5.1: Open-access datasets for RSS-based indoor positioning

Dataset	Year	Description
UJIIndoorLoc [92] [93]	2014	<ul style="list-style-type: none"> • Use Wi-Fi technology. • Use 520 access points and 25 mobile devices. • Data were collected by 20 users. • Three multi-floor buildings with 108,703 m² total area. Each building has 4 to 5 floors. • Dataset contains 21,049 fingerprint samples. • Testing data were collected 4 months after training data. At least 60% of training data and testing data were collected within one day. The rest were collected within 14 days and 20 days for training data and testing data respectively. • Coordinates of data collection points do not repeat.
Crowdsourced Wi-Fi dataset [94] [95]	2017	<ul style="list-style-type: none"> • Use Wi-Fi technology. • Use 991 access points. • Data were collected by 8 users. • Five-floor single-building with 22,570 m² footprint. • Dataset contains 4,648 fingerprints collected within 8 months, randomly split into 697 training data and 3,951 testing data. • Coordinates of data collection points do not repeat.
BLE RSSI dataset [96] [97]	2018	<ul style="list-style-type: none"> • Use iBeacon (BLE) technology. • Use 13 iBeacons and 1 mobile device. • The number of users is not specified. • Single-floor single-building with size of 60 x 54 m. • Dataset contains 6,620 fingerprint samples collected within 7 months, which consists of 820 labeled data for training, 600 labeled data points for testing, and 5200 unlabeled data points for semi-supervised learning. • Coordinates of data collection points do repeat.
Long-term Wi-Fi dataset [98] [99]	2018	<ul style="list-style-type: none"> • Use Wi-Fi technology. • Use 576 access points and 1 mobile device. • Data were collected by 1 user. • Two-floor single-building with 308.4 m² total area. • Contains 63,504 fingerprint samples collected within 15 months. • Coordinates of data collection points do repeat. • At least 1 training campaign and 5 testing campaigns were conducted every month. For the first month, there are 15 training campaigns. • For testing campaign 1 and 5, data were collected at the same coordinates used for training data. For testing campaign 2, 3, and 4, data were collected at different coordinates.

5.3 Dataset description and data adaptation

Long-term Wi-Fi dataset [98] [99] contains training and testing fingerprints collected every month during 15 months. There are 15 training campaigns and 5 testing campaigns on the first month, and 1 training campaign and 5 testing campaigns on each of the subsequent months. On each campaign, training fingerprints were recorded at 48 locations in total, distributed over two floors with floor heights of 3 m and 5 m respectively. The 24 training points on each floor are as shown in Figure 5.1, which are identical for both floor. We processed fingerprints from both floors altogether and use 3-dimensional representation of the location coordinates.

Figure 5.1 also shows the coordinates the reference points at the four corners of each floor. Since there are 2 floors, we have 8 reference points in total. Those points are selected to replace the fingerprints supposedly coming from reference devices.

In our performance evaluation, we used 5 testing campaigns available for each month in the dataset, comprising 2600 testing fingerprints in total for each month. Among the testing campaigns, the tracking device location coordinates at testing campaign #1 and #5 are identical with the location coordinates at the training campaigns, whereas testing campaign #2, #3, and #4 have different coordinates.

Figure 5.1: Site layout of long-term Wi-Fi dataset

The selection of training and testing campaigns as training and testing data/reference is presented in Figure 5.2 and described as follows:

1. Out of the 15 training campaigns available on the 1st month, we only used 2 most recent campaigns, which is training campaign #14 and #15, as training fingerprints. Reference fingerprints at training phase are drawn from those training fingerprints at the reference points.
2. We used the fingerprints of reference points from the training campaign on the testing month as additional training fingerprints and reference fingerprints at testing phase. Since our proposed methods are derived from LEMT and designed to estimate those fingerprints using functional relationship model in Eq. (3.7), the remaining points on the testing month can be directly estimated from the reference points. In a practical scenario,

this means that, on the testing months, we only need repeat the data collection at reference points only, which is more efficient than conventional approach where data collection must be exhaustively repeated for all training points.

3. Testing campaign #1, #2, #3, #4, and #5 are all used as testing data.

Figure 5.2: Configuration of training and testing data/reference for long-term Wi-Fi dataset

Across the campaigns, there are 448 access points in total, appearing in and out each month. RSS value is represented in dBm, ranging from -100 to -31. In the original dataset, RSS of 100 is used to represent undetected access points in the original dataset. In our dataset, however, we use RSS of -120 to represent

undetected access points since it is more reasonable to assume that undetected access points are the ones having low RSS, as if the signal is loss, rather than high RSS. In our location fingerprinting, the Wi-Fi access points are treated as anchors.

To lower the computational complexity, we excluded anchors with too many undetected RSS in the training campaign. An anchor must have at least 30% valid RSS in the training campaign to be considered a valid anchor that we include in the computation. This percentage can be adjusted higher or lower, which will affect the training time and testing time.

In Wi-Fi based location fingerprinting datasets, including the long-term Wi-Fi dataset, coordinates of the access points are typically not provided. However, the location fingerprinting methods we discussed in this thesis do not need the coordinates of anchors, thus the absence of those coordinates does not affect the usability of the dataset.

Unlike in the previous chapters, in this chapter we excluded LANDMARC from the benchmark methods since the method solely relies on reference devices. When fingerprints from reference devices are replaced with training fingerprints, LANDMARC would be completely reduced into NN.

5.4 Performance evaluation

5.4.1 Accuracy

The accuracy results presented in Table 5.2 and Figure 5.3 show that FCO-LEELM has the lowest error, followed by CO-LEELM and CO-LEMT. From the 1st month to 10th month, the accuracies of all methods are relatively stable. In line with the results found by Mendoza-Silva et al. [98], we found that NN method also experiences a significant error increase on the 11th month. This increase is likely to be caused by temporal variation, where the variation in the fingerprint characteristics of the 11th month and beyond is quite significant as compared to the prior months.

However, unlike the results found by Mendoza-Silva et al. [98] where the error recovers on the 12th month and beyond, the error we obtained for those months remain high. This difference is understandable since our experiment setup only use the fingerprints of 8 reference points located at the four corners of each floor to update the training fingerprints on the given month, whereas the original setup uses the whole training fingerprints collected at 48 locations [98].

The errors of our proposed FCO-LEELM and CO-LEELM on the 12th and beyond, on the other hand, recover quite well. This indicates that the functional relationship model in the proposed methods can estimate the fingerprints at the 48 training locations accurately, although we only use 8 reference points to update the system each month.

Table 5.2: Accuracy of long-term Wi-Fi dataset

Month index	NN	LEMT	CO-LEMT	LEELM	CO-LEELM	F-LEELM	FCO-LEELM
	90th error percentile in meters (mean $\pm$ standard deviation)						
1	5.39 $\pm$ 0.00	5.68 $\pm$ 0.00	5.18 $\pm$ 0.00	5.41 $\pm$ 0.05	4.70 $\pm$ 0.02	5.39 $\pm$ 0.00	4.59 $\pm$ 0.01
2	5.39 $\pm$ 0.00	5.68 $\pm$ 0.00	4.66 $\pm$ 0.00	5.68 $\pm$ 0.01	4.69 $\pm$ 0.02	5.39 $\pm$ 0.00	4.57 $\pm$ 0.01
3	5.68 $\pm$ 0.00	6.35 $\pm$ 0.00	5.26 $\pm$ 0.00	5.71 $\pm$ 0.06	5.22 $\pm$ 0.08	5.68 $\pm$ 0.00	4.90 $\pm$ 0.02
4	5.68 $\pm$ 0.00	6.64 $\pm$ 0.00	5.39 $\pm$ 0.00	5.68 $\pm$ 0.00	4.88 $\pm$ 0.04	5.68 $\pm$ 0.00	4.77 $\pm$ 0.02
5	5.66 $\pm$ 0.00	6.47 $\pm$ 0.00	5.04 $\pm$ 0.00	5.67 $\pm$ 0.01	4.78 $\pm$ 0.04	5.66 $\pm$ 0.00	4.60 $\pm$ 0.01
6	5.66 $\pm$ 0.00	5.68 $\pm$ 0.00	4.92 $\pm$ 0.00	5.62 $\pm$ 0.09	4.85 $\pm$ 0.05	5.64 $\pm$ 0.06	4.68 $\pm$ 0.01
7	5.39 $\pm$ 0.00	5.68 $\pm$ 0.00	4.96 $\pm$ 0.00	5.67 $\pm$ 0.01	4.58 $\pm$ 0.02	5.48 $\pm$ 0.06	4.53 $\pm$ 0.02
8	5.39 $\pm$ 0.00	6.63 $\pm$ 0.00	4.80 $\pm$ 0.00	5.44 $\pm$ 0.03	4.59 $\pm$ 0.02	5.39 $\pm$ 0.00	4.48 $\pm$ 0.01
9	5.68 $\pm$ 0.00	5.80 $\pm$ 0.00	5.18 $\pm$ 0.00	6.18 $\pm$ 0.33	4.81 $\pm$ 0.03	5.68 $\pm$ 0.00	4.64 $\pm$ 0.01
10	6.47 $\pm$ 0.00	6.58 $\pm$ 0.00	5.13 $\pm$ 0.00	5.91 $\pm$ 0.27	4.87 $\pm$ 0.06	6.20 $\pm$ 0.28	4.77 $\pm$ 0.01
11	7.48 $\pm$ 0.00	8.39 $\pm$ 0.00	7.03 $\pm$ 0.00	7.03 $\pm$ 0.12	6.57 $\pm$ 0.05	7.11 $\pm$ 0.00	6.65 $\pm$ 0.01
12	8.34 $\pm$ 0.00	12.56 $\pm$ 0.00	8.40 $\pm$ 0.00	9.02 $\pm$ 0.30	5.22 $\pm$ 0.12	8.32 $\pm$ 0.06	5.05 $\pm$ 0.02
13	7.96 $\pm$ 0.00	11.59 $\pm$ 0.00	7.93 $\pm$ 0.00	8.86 $\pm$ 0.18	5.18 $\pm$ 0.19	7.95 $\pm$ 0.02	5.34 $\pm$ 0.02
14	7.96 $\pm$ 0.00	11.32 $\pm$ 0.00	6.68 $\pm$ 0.00	9.11 $\pm$ 0.26	5.16 $\pm$ 0.17	7.93 $\pm$ 0.02	4.94 $\pm$ 0.03
15	7.98 $\pm$ 0.00	11.25 $\pm$ 0.00	6.69 $\pm$ 0.00	8.78 $\pm$ 0.22	5.17 $\pm$ 0.18	7.96 $\pm$ 0.03	5.07 $\pm$ 0.02
	95th error percentile in meters (mean $\pm$ standard deviation)						
1	6.58 $\pm$ 0.00	7.78 $\pm$ 0.00	6.21 $\pm$ 0.00	6.63 $\pm$ 0.02	5.71 $\pm$ 0.06	5.78 $\pm$ 0.04	5.57 $\pm$ 0.01
2	6.63 $\pm$ 0.00	6.82 $\pm$ 0.00	5.61 $\pm$ 0.00	6.73 $\pm$ 0.12	5.71 $\pm$ 0.03	6.63 $\pm$ 0.01	5.59 $\pm$ 0.01
3	7.48 $\pm$ 0.00	7.87 $\pm$ 0.00	5.97 $\pm$ 0.00	7.62 $\pm$ 0.13	5.91 $\pm$ 0.04	7.00 $\pm$ 0.15	5.76 $\pm$ 0.00
4	7.11 $\pm$ 0.00	8.16 $\pm$ 0.00	6.63 $\pm$ 0.00	7.12 $\pm$ 0.22	5.81 $\pm$ 0.02	6.92 $\pm$ 0.04	5.74 $\pm$ 0.02
5	6.64 $\pm$ 0.00	7.98 $\pm$ 0.00	6.45 $\pm$ 0.00	6.90 $\pm$ 0.26	5.85 $\pm$ 0.06	6.64 $\pm$ 0.00	5.65 $\pm$ 0.02
6	7.02 $\pm$ 0.00	7.78 $\pm$ 0.00	6.16 $\pm$ 0.00	6.88 $\pm$ 0.17	5.92 $\pm$ 0.05	7.07 $\pm$ 0.07	5.81 $\pm$ 0.01
7	6.53 $\pm$ 0.00	7.11 $\pm$ 0.00	6.12 $\pm$ 0.00	6.64 $\pm$ 0.01	5.74 $\pm$ 0.03	6.58 $\pm$ 0.00	5.65 $\pm$ 0.04
8	6.47 $\pm$ 0.00	7.96 $\pm$ 0.00	5.84 $\pm$ 0.00	6.64 $\pm$ 0.01	5.60 $\pm$ 0.03	6.63 $\pm$ 0.00	5.48 $\pm$ 0.02
9	7.48 $\pm$ 0.00	7.78 $\pm$ 0.00	6.20 $\pm$ 0.00	7.75 $\pm$ 0.09	5.90 $\pm$ 0.05	7.04 $\pm$ 0.08	5.72 $\pm$ 0.02
10	7.78 $\pm$ 0.00	7.78 $\pm$ 0.00	5.99 $\pm$ 0.00	7.78 $\pm$ 0.08	5.90 $\pm$ 0.04	7.78 $\pm$ 0.00	5.84 $\pm$ 0.02
11	8.49 $\pm$ 0.00	9.45 $\pm$ 0.00	7.87 $\pm$ 0.00	8.09 $\pm$ 0.20	7.38 $\pm$ 0.12	8.47 $\pm$ 0.04	7.61 $\pm$ 0.05
12	9.45 $\pm$ 0.00	13.59 $\pm$ 0.00	9.17 $\pm$ 0.00	9.98 $\pm$ 0.29	5.80 $\pm$ 0.13	9.28 $\pm$ 0.10	5.65 $\pm$ 0.01
13	8.96 $\pm$ 0.00	12.71 $\pm$ 0.00	8.71 $\pm$ 0.00	9.73 $\pm$ 0.20	5.73 $\pm$ 0.22	8.96 $\pm$ 0.00	6.03 $\pm$ 0.02
14	9.00 $\pm$ 0.00	12.53 $\pm$ 0.00	7.34 $\pm$ 0.00	9.98 $\pm$ 0.27	5.77 $\pm$ 0.21	8.96 $\pm$ 0.02	5.63 $\pm$ 0.01
15	9.33 $\pm$ 0.00	12.53 $\pm$ 0.00	7.39 $\pm$ 0.00	9.75 $\pm$ 0.28	5.80 $\pm$ 0.15	8.96 $\pm$ 0.01	5.71 $\pm$ 0.01

(a) Percentile 90

(b) Percentile 95

Figure 5.3: Accuracy of long-term Wi-Fi dataset

A clearer comparison can be observed in Figure 5.4, where FCO-LEELM and CO-LEELM consistently produce lowest error across all months. On the 1st and 11th month, the two proposed methods are comparable with CO-LEMT, and slightly better than other methods. On the 12th and 15th month, the gaps between the two proposed methods and other methods become larger. This result indicates that the use of ELM as functional relationship model might provide better generalization than M5 model tree originally used in the LEMT method.

In all months, continuous-output methods always produce lower error than their equivalent discrete-output methods. This phenomenon is expected since the testing campaigns in the dataset do not completely intersect with the training campaigns.

Figure 5.4: Cumulative error distribution of long-term Wi-Fi dataset

5.4.2 Floor detection

We analyzed the success rate of the floor detection by comparing the estimated floor height and the ground-truth floor height. Since continuous-output methods would produce continuous values on the floor height, we round the estimated floor heights to the nearest decimal before making the comparison.

Despite their better error percentile, Figure 5.5 shows that continuous-output methods have lower success rate when it comes to floor detection, particularly

for the 12th month and beyond. NN method has a good success rate in floor detection, which is consistent with the result reported in the original work [98].

Figure 5.5: Floor detection success rate of long-term Wi-Fi dataset

The lower success rate in floor detection could be caused by the continuous-output methodology, where the output coordinate is obtained from a combination of several coordinates having high similarity. When those coordinates come from different floors and combined into a 3D coordinate, we might have an ambiguous floor height which is neither 3 nor 5, which are the only possible floor heights of the ground-truth testing coordinates in the dataset under test. Further analysis and a combination of continuous-output and discrete-output would be helpful to improve the floor detection success rate while keeping the overall error percentile low.

5.4.3 Training time and testing time

Table 5.3 shows training time and testing time we obtained for different months. In general, the results are similar with the results we obtained in Chapter 3 and Chapter 4. Although the processing time of long-term Wi-Fi dataset is much longer than the result we had in the previous chapters, the longer processing time is a natural implication of having a bigger dataset with more samples of both training fingerprints and testing fingerprints.

Similar with the results we obtained in the previous chapters, the choice between discrete output and continuous output does not have noticeable impact on the training time or testing time as much as the choice between imputation and filtering does. Also, the use of ELM significantly lowers the training time and testing time, when compared to M5 model tree used in LEMT.

Among the benchmark methods, NN has the highest speed and LEMT has the lowest speed. Among our proposed methods, LEELM and CO-LEELM are the fastest, whereas CO-LEMT is the slowest. The training time and testing time of the filtered ELM methods, F-LEELM and FCO-LEELM, are about 2 times longer than those of LEELM and CO-LEELM.

Considering the speed difference and the results showing that imputation and filtering do not appear to have significant impact on the accuracy, it is more beneficial to choose imputation over filtering to deal with incomplete data in location fingerprinting system.

Table 5.3: Training time and testing time of long-term Wi-Fi dataset

Month index	NN	LEMT	CO-LEMT	LEELM	CO-LEELM	F-LEELM	FCO-LEELM
	Training time in seconds (mean $\pm$ standard deviation)						
1	-	2.12 $\pm$ 0.36	2.01 $\pm$ 0.19	0.35 $\pm$ 0.03	0.35 $\pm$ 0.03	0.59 $\pm$ 0.05	0.60 $\pm$ 0.05
2	-	1.87 $\pm$ 0.03	1.87 $\pm$ 0.04	0.33 $\pm$ 0.01	0.32 $\pm$ 0.01	0.58 $\pm$ 0.01	0.58 $\pm$ 0.01
3	-	1.87 $\pm$ 0.02	1.90 $\pm$ 0.09	0.33 $\pm$ 0.02	0.32 $\pm$ 0.00	0.57 $\pm$ 0.01	0.57 $\pm$ 0.01
4	-	1.88 $\pm$ 0.09	1.35 $\pm$ 0.02	0.33 $\pm$ 0.01	0.32 $\pm$ 0.01	0.58 $\pm$ 0.01	0.57 $\pm$ 0.01
5	-	1.88 $\pm$ 0.02	1.87 $\pm$ 0.02	0.33 $\pm$ 0.01	0.32 $\pm$ 0.01	0.56 $\pm$ 0.01	0.56 $\pm$ 0.01
6	-	1.87 $\pm$ 0.02	1.87 $\pm$ 0.02	0.33 $\pm$ 0.02	0.32 $\pm$ 0.00	0.57 $\pm$ 0.01	0.57 $\pm$ 0.01
7	-	1.89 $\pm$ 0.04	1.88 $\pm$ 0.02	0.33 $\pm$ 0.02	0.32 $\pm$ 0.01	0.57 $\pm$ 0.01	0.56 $\pm$ 0.01
8	-	1.87 $\pm$ 0.02	1.87 $\pm$ 0.01	0.33 $\pm$ 0.01	0.33 $\pm$ 0.00	0.57 $\pm$ 0.01	0.58 $\pm$ 0.01
9	-	1.87 $\pm$ 0.01	1.88 $\pm$ 0.02	0.33 $\pm$ 0.02	0.32 $\pm$ 0.01	0.57 $\pm$ 0.01	0.57 $\pm$ 0.01
10	-	1.87 $\pm$ 0.01	1.88 $\pm$ 0.01	0.33 $\pm$ 0.01	0.32 $\pm$ 0.01	0.57 $\pm$ 0.01	0.57 $\pm$ 0.01
11	-	1.88 $\pm$ 0.02	1.87 $\pm$ 0.01	0.33 $\pm$ 0.02	0.33 $\pm$ 0.00	0.57 $\pm$ 0.01	0.57 $\pm$ 0.01
12	-	1.88 $\pm$ 0.01	1.88 $\pm$ 0.02	0.32 $\pm$ 0.00	0.32 $\pm$ 0.00	0.52 $\pm$ 0.01	0.53 $\pm$ 0.01
13	-	1.87 $\pm$ 0.01	1.88 $\pm$ 0.01	0.33 $\pm$ 0.02	0.32 $\pm$ 0.01	0.52 $\pm$ 0.01	0.53 $\pm$ 0.01
14	-	1.92 $\pm$ 0.05	1.93 $\pm$ 0.08	0.33 $\pm$ 0.02	0.33 $\pm$ 0.01	0.54 $\pm$ 0.02	0.54 $\pm$ 0.03
15	-	1.89 $\pm$ 0.04	1.89 $\pm$ 0.05	0.33 $\pm$ 0.01	0.33 $\pm$ 0.00	0.52 $\pm$ 0.01	0.53 $\pm$ 0.03
	Testing time in seconds (mean $\pm$ standard deviation)						
1	0.03 $\pm$ 0.01	28.44 $\pm$ 2.60	27.92 $\pm$ 2.48	3.62 $\pm$ 0.33	3.47 $\pm$ 0.22	8.16 $\pm$ 0.58	8.10 $\pm$ 0.50
2	0.03 $\pm$ 0.00	27.03 $\pm$ 0.07	27.00 $\pm$ 0.12	3.33 $\pm$ 0.02	3.21 $\pm$ 0.02	7.82 $\pm$ 0.03	7.80 $\pm$ 0.09
3	0.03 $\pm$ 0.00	27.35 $\pm$ 0.93	26.92 $\pm$ 0.25	3.33 $\pm$ 0.02	3.26 $\pm$ 0.06	7.65 $\pm$ 0.05	7.58 $\pm$ 0.03
4	0.03 $\pm$ 0.00	26.88 $\pm$ 0.09	26.80 $\pm$ 0.05	3.34 $\pm$ 0.04	3.21 $\pm$ 0.03	7.60 $\pm$ 0.08	7.54 $\pm$ 0.05
5	0.03 $\pm$ 0.00	26.88 $\pm$ 0.10	26.93 $\pm$ 0.12	3.32 $\pm$ 0.04	3.23 $\pm$ 0.03	7.49 $\pm$ 0.09	7.44 $\pm$ 0.04
6	0.03 $\pm$ 0.00	27.07 $\pm$ 0.11	26.99 $\pm$ 0.06	3.36 $\pm$ 0.06	3.22 $\pm$ 0.04	7.52 $\pm$ 0.02	7.49 $\pm$ 0.05
7	0.03 $\pm$ 0.00	26.98 $\pm$ 0.08	26.91 $\pm$ 0.16	3.32 $\pm$ 0.03	3.20 $\pm$ 0.03	7.54 $\pm$ 0.02	7.49 $\pm$ 0.04
8	0.03 $\pm$ 0.00	26.97 $\pm$ 0.11	26.86 $\pm$ 0.06	3.31 $\pm$ 0.02	3.23 $\pm$ 0.08	7.64 $\pm$ 0.04	7.57 $\pm$ 0.03
9	0.03 $\pm$ 0.00	27.14 $\pm$ 0.14	27.09 $\pm$ 0.10	3.41 $\pm$ 0.21	3.23 $\pm$ 0.07	7.59 $\pm$ 0.09	7.52 $\pm$ 0.08
10	0.03 $\pm$ 0.00	27.46 $\pm$ 0.08	27.28 $\pm$ 0.15	3.32 $\pm$ 0.03	3.21 $\pm$ 0.02	7.52 $\pm$ 0.03	7.48 $\pm$ 0.06
11	0.03 $\pm$ 0.00	26.98 $\pm$ 0.05	26.97 $\pm$ 0.08	3.33 $\pm$ 0.04	3.24 $\pm$ 0.06	6.88 $\pm$ 0.03	6.81 $\pm$ 0.04
12	0.03 $\pm$ 0.00	26.86 $\pm$ 0.07	26.81 $\pm$ 0.12	3.35 $\pm$ 0.03	3.25 $\pm$ 0.04	6.16 $\pm$ 0.02	6.07 $\pm$ 0.03
13	0.03 $\pm$ 0.00	27.00 $\pm$ 0.06	27.03 $\pm$ 0.08	3.35 $\pm$ 0.03	3.26 $\pm$ 0.04	6.15 $\pm$ 0.03	6.07 $\pm$ 0.04
14	0.03 $\pm$ 0.00	27.45 $\pm$ 0.36	27.75 $\pm$ 1.06	3.40 $\pm$ 0.05	3.31 $\pm$ 0.05	6.41 $\pm$ 0.23	6.32 $\pm$ 0.30
15	0.03 $\pm$ 0.00	27.15 $\pm$ 0.16	27.03 $\pm$ 0.15	3.34 $\pm$ 0.03	3.24 $\pm$ 0.03	6.21 $\pm$ 0.03	6.13 $\pm$ 0.05

5.5 Summary

In this chapter, we have presented an evaluation of our proposed fingerprinting methods using an open-access long-term Wi-Fi dataset, which does not contain static reference devices. Although our proposed methods originally need reference devices, we can replace those data with the fingerprints collected using the tracking device on the testing month.

Unlike the conventional data collection, we do not need to repeat data collection in the testing month as exhaustively as in the initial training month. We observed that good accuracy can be achieved even though the repeated data collection only consider 8 reference points each month, out of the 48 training points originally recorded on the 1st month. The lower number of collection points during the repeated data collection allows us to have the repeated data collection process less tedious. Our experiment on the long-term Wi-Fi dataset shows that the proposed CO-LEELM and FCO-LEELM can maintain good accuracy across 15 months.

However, there are still room for improvement on CO-LEELM and FCO-LEELM, particularly because their floor detection success rates are relatively unstable for the 12th month and beyond as compared to the discrete-output methods.

As for the training time and testing time, CO-LEELM that uses imputation method to deal with incomplete fingerprints has lower training time and testing time than FCO-LEELM that uses filtering. Since their accuracies are about the same, CO-LEELM is preferred due to its shorter training time and testing time.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Indoor positioning system is a key technology for future applications that involves navigation, automation, and optimization. Various devices and computational methods for indoor positioning have been developed in the past two decades, yet there are still many unresolved challenges that hinder them from being applied in today's market. Those include limited accuracy, lack of robustness over temporal and device variation, high system cost, long processing time, and repetitive recalibrations. This thesis presents machine learning approach for location fingerprinting that address some of those problems.

Based on some existing location fingerprinting methods, which are NN, LANDMARC, and LEMT, we introduced novel location fingerprinting methods aimed to achieve better accuracy, shorter training and testing time, and better robustness against reduced number of training points and reference devices. The methods combine the existing location fingerprinting methods and involve machine learning approach.

In Chapter 3, we combined the methodology of LEMT and LANDMARC to produce a continuous-output location fingerprinting method CO-LEMT. The

proposed CO-LEMT achieves good accuracy when the training points are not as densely distributed as the testing points. Using 3 datasets with 4 scenarios, we found that CO-LEMT method produces 0.5 m to 2 m accuracy improvement as compared to the existing location fingerprinting methods. The method, however, requires longer training time and testing time as compared to NN and LANDMARC it is developed based on LEMT that requires long training time and testing time.

To address the long training time and testing time of CO-LEMT, we proposed the use of ELM for location fingerprinting in Chapter 4. We developed a framework to translate location fingerprinting methods into ELM-based techniques and applied it on the proposed CO-LEMT method. We found that incomplete data can be handled either through imputation or filtering. Both handling methods have similar accuracy, but CO-LEELM method that uses imputation has shorter training time and testing time. In general, the speed of the proposed CO-LEELM is about 6 to 10 times faster and 15 to 30 times faster than CO-LEMT in terms of training time and testing time, respectively. However, for all datasets and test scenarios under test, temporal variation is found to cause a quite significant accuracy degradation for all location fingerprinting methods, including our proposed method.

In Chapter 5, we evaluated the proposed location fingerprinting methods using an open-access long-term Wi-Fi dataset that contains no reference device. Instead of exhaustively repeating the training fingerprints recording every month, we

proposed the use of fingerprints at selected reference points to allow the methods to capture temporal variation. Even with the limited amount of fingerprint updates, our proposed CO-LEELM produces error percentiles that are consistently lower than NN and LEMT across the months. However, we observed that the proposed method produces lower success rate in detecting the floor height, and thus further improvement on this aspect is required.

6.2 Recommendations for future work

Based on the study we have conducted, robustness to temporal variation problem is one of the areas that still need further improvement. Although our proposed method produces better accuracy than the existing techniques, the positioning error under spatial-temporal variation remains much higher than the positioning error under spatial variation. Without good robustness to temporal variation, we need to repeat the manual collection of training data periodically to ensure that the training data remain relevant with the latest environmental condition. This is not favorable considering the manpower requirement for data collection.

One possible way to achieve better robustness to temporal variation is through a fusion approach that involves other sensors. However, accuracy should not be the only consideration when building the fusion approach. Current research works involving motion sensors or vision sensors rely on continuous filtering and tracking of sensor data, which is highly complex and power consuming. We

believe that greater effort should be directed towards developing computationally-efficient methods that can be implemented on portable devices for real-time applications.

Currently, the placements of anchors and reference devices are determined intuitively by considering the geometrical shape of the test-site. Anchors and reference points are placed at the corners of the rooms or arranged in a grid separated by some distance spacing. While this might be considered adequate in current practices, there could be a better way to find an optimal placement of anchors and reference devices. Such placement might be determined in an automated manner and derived through machine learning approach, with an objective to produce the best trade-off between cost and accuracy.

When searching for an open-access dataset to test our proposed methods, we also found that RSS-based indoor positioning datasets are still very limited, both in quantity and quality. Although there have been some efforts in publishing datasets in the past 4 years, the datasets are only suitable for a limited class of indoor positioning methods due to the methodology of the data collection. We found that most datasets contain different training coordinates at each campaign, except for the long-term Wi-Fi dataset. Data collection methodologies in other datasets do not consider that some methods may need to be updated with fingerprints recorded at the same training coordinates each time. Also, to the best of our knowledge, dataset that uses of reference devices is still nowhere to be found. Although we attempted to generate our own datasets to fulfill the needs of

our proposed methods, the scale of our data collections is very limited due to time and resource limitation. A large-scale project that considers the unique needs of different indoor positioning methods, while capturing the characteristics of spatial variation in large buildings and the characteristics of temporal variation in long-term duration, would be beneficial to advance the progress of indoor positioning research.

Author's Publications

Journal papers

1. F. Dwiyaasa, M.-H. Lim, Y.-S. Ong, and B. Panigrahi, "Extreme learning machine for indoor location fingerprinting," in *Multidimensional Systems and Signal Processing*, vol. 28, no. 3, pp. 867-883, 2017.

Conference papers

1. F. Dwiyaasa and M.-H. Lim, "Zone classification for low-power active RFID tags," in *Proceedings of IEEE International Symposium on Integrated Circuits (ISIC)*, 2014, pp. 572-575.
2. F. Dwiyaasa and M.-H. Lim, "Extreme Learning Machine for active RFID location classification," in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 2015, pp. 657-670.
3. F. Dwiyaasa and M.-H. Lim, "Identifying Indistinguishable Classes in Multi-class Classification Data Sets Using ELM," in *Proceedings of International Conference on Extreme Learning Machine (ELM) 2014*, vol. 1, 2015, pp. 407-415.
4. F. Dwiyaasa, M.-H. Lim, Y.-S. Ong, and B. Panigrahi, "Equality constrained-optimization-based semi-supervised ELM for modeling signal strength temporal variation in indoor location estimation," in *Proceedings of International Conference on Extreme Learning Machine (ELM) 2015*, vol. 1, 2016, pp. 383-395.
5. F. Dwiyaasa and M.-H. Lim, "A survey of problems and approaches in wireless-based indoor positioning," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1-7.

Bibliography

- [1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067-1080, 2007.
- [2] M. Youssef and A. Agrawala, "Small-scale compensation for WLAN location determination systems," in *Proceedings of IEEE Wireless Communications and Networking Conference*, 2003, vol. 3, pp. 1974-1978.
- [3] G. Çaliş, B. Becerik-Gerber, and A. B. Göktepe, "Analysis of the variability of RSSI values for active RFID-based indoor applications," *Turkish Journal of Engineering and Environmental Sciences*, vol. 37, no. 2, pp. 186-210, 2013.
- [4] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000, vol. 2, pp. 775-784.
- [5] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2004, pp. 14-23.
- [6] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 869-883, 2008.
- [7] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization," in *Proceedings of IEEE International Conference on Localization and GNSS (ICL-GNSS)*, 2011, pp. 53-57.
- [8] A. W. Tsui, Y.-H. Chuang, and H.-H. Chu, "Unsupervised learning for solving RSS hardware variance problem in WiFi localization," *Mobile Networks and Applications*, vol. 14, no. 5, pp. 677-691, 2009.
- [9] D. Seetharam and R. Fletcher, "Battery-powered RFID," in *Proceedings of the 1st ACM Workshop on Convergence of RFID and Wireless Sensor Networks and their Applications (SenseID)*, 2007.
- [10] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless networks*, vol. 10, no. 6, pp. 701-710, 2004.
- [11] R. Oktem, E. U. Aydin, and N. E. Cagiltay, "An RFID based location finding and tracking with guidance," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and*

- Mobile Computing (WiCOM)*, 2008, pp. 1-4.
- [12] T. Kämpke, B. Kluge, E. Prassler, and M. Strobel, "Robot position estimation on a RFID-tagged smart floor," in *Field and Service Robotics, Springer Tracts in Advanced Robotics*, vol. 42, 2008, pp. 201-211.
 - [13] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall, 2001.
 - [14] A. Hatami and K. Pahlavan, "Comparative statistical analysis of indoor positioning using empirical data and indoor radio channel models," in *Proceedings of IEEE Consumer Communications and Networking Conference*, 2006, vol. 2, pp. 1018-1022.
 - [15] M. Emery and M. K. Denko, "IEEE 802.11 WLAN based real-time location tracking in indoor and outdoor environments," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, 2007, pp. 1062-1065.
 - [16] *Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 MHz to 100 GHz*, 2012.
 - [17] C.-F. Yang, B.-C. Wu, and C.-J. Ko, "A ray-tracing method for modeling indoor wave propagation and penetration," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 6, pp. 907-919, Jun 1998.
 - [18] K. El-Kafrawy, M. Youssef, A. El-Keyi, and A. Naguib, "Propagation modeling for accurate indoor WLAN RSS-based localization," in *Proceedings of the 72nd IEEE Vehicular Technology Conference Fall (VTC-Fall)*, 2010, pp. 1-5.
 - [19] G. Kloos, "Radio-frequency signal strength based localisation in unstructured outdoor environments," Australian Centre for Field Robotics, Department of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, 2007.
 - [20] Y. Chen and H. Kobayashi, "Signal strength based indoor geolocation," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2002, vol. 5, pp. 436-439.
 - [21] M. Youssef and A. Agrawala, "Handling samples correlation in the Horus system," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004, vol. 2, pp. 1023-1031.
 - [22] H. Lim, L. C. Kung, J. C. Hou, and H. Luo, "Zero-Configuration, Robust Indoor Localization: Theory and Experimentation," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006, pp. 1-12.
 - [23] S.-P. Kuo and Y.-C. Tseng, "A scrambling method for fingerprint positioning based on temporal diversity and spatial dependency," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 678-684, 2008.
 - [24] S.-H. Fang, T.-N. Lin, and K.-C. Lee, "A novel algorithm for multipath fingerprinting in indoor WLAN environments," *IEEE transactions on wireless communications*, vol. 7, no. 9, 2008.

- [25] S.-H. Fang, C.-H. Wang, S.-M. Chiou, and P. Lin, "Calibration-free approaches for robust Wi-Fi positioning against device diversity: A performance comparison," in *Proceedings of the 75th IEEE International Conference on Vehicular Technology Conference (VTC Spring)*, 2012, pp. 1-5.
- [26] Z.-H. Wu, Y. Han, Y. Chen, and K. J. R. Liu, "A time-reversal paradigm for indoor positioning system," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1331-1339, 2015.
- [27] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763-776, 2017.
- [28] C. Nerguizian, C. Despins, and S. Affès, "Geolocation in mines with an impulse response fingerprinting technique and neural networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 3, pp. 603-611, 2006.
- [29] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, p. 25, 2013.
- [30] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2015, pp. 1666-1671.
- [31] S. Lanzisera, D. T. Lin, and K. S. J. Pister, "RF time of flight ranging for wireless sensor network localization," in *Proceedings of IEEE International Workshop on Intelligent Solutions in Embedded Systems*, 2006, pp. 1-12.
- [32] J. Heidrich, D. Brenk, J. Essel, G. Fischer, R. Weigel, and S. Schwarzer, "Local positioning with passive UHF RFID transponders," in *Proceedings of IEEE MTT-S International Microwave Workshop on Wireless Sensing, Local Positioning, and RFID*, 2009.
- [33] S. Schwarzer, M. Vossiek, M. Pichler, and A. Stelzer, "Precise distance measurement with IEEE 802.15. 4 (ZigBee) devices," in *Proceedings of IEEE Radio and Wireless Symposium*, 2008, pp. 779-782.
- [34] X. Li and K. Pahlavan, "Super-resolution TOA estimation with diversity for indoor geolocation," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 224-234, 2004.
- [35] M. Heidari, F. O. Akgul, N. A. Alsindi, and K. Pahlavan, "Neural network assisted identification of the absence of direct path in indoor localization," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, 2007, pp. 387-392.
- [36] L. Yu, M. Laaraiedh, S. Avrillon, and B. Uguen, "Fingerprinting localization based on neural networks and ultra-wideband signals," in *Proceedings of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2011, pp. 184-189.
- [37] D. Niculescu and B. Nath, "VOR base stations for indoor 802.11

- positioning," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, pp. 58-69.
- [38] A. Cidronali, S. Maddio, G. Giorgetti, and G. Manes, "Analysis and performance of a smart antenna for 2.45-GHz single-anchor indoor positioning," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 1, pp. 21-31, 2010.
 - [39] C. Wong, R. Klukas, and G. G. Messier, "Using WLAN infrastructure for angle-of-arrival indoor user location," in *Proceedings of IEEE 68th Vehicular Technology Conference (VTC)*, 2008, pp. 1-5.
 - [40] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181-1186, 2010.
 - [41] Y. Zhang, X. Li, and M. Amin, "Principles and techniques of RFID positioning," *{RFID} systems: Research trend and challenges*, pp. 389-415, 2010.
 - [42] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005, pp. 91-98.
 - [43] X. Li, B. Hua, Y. Shang, Y. Guo, and L. Yue, "Bilateration: an attack-resistant localization algorithm of wireless sensor network," *Embedded and Ubiquitous Computing*, pp. 321-332, 2007.
 - [44] J. Cota-Ruiz, J.-G. Rosiles, E. Sifuentes, and P. Rivas-Perea, "A low-complexity geometric bilateration method for localization in wireless sensor networks and its comparison with least-squares methods," *Sensors*, vol. 12, no. 1, pp. 839-862, 2012.
 - [45] C. C. Ho and R. Lee, "Real-Time Indoor Positioning System Based on RFID Heron-Bilateration Location Estimation and IMU Inertial-Navigation Location Estimation," in *Proceedings of the 39th IEEE Computer Software and Applications Conference (COMPSAC)*, 2015, vol. 3, pp. 481-486.
 - [46] H. Zou, H. Wang, L. Xie, and Q.-S. Jia, "An RFID indoor positioning system by using weighted path loss and extreme learning machine," in *Proceedings of the 1st IEEE International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2013, pp. 66-71.
 - [47] M. Gunawan, B. Li, T. Gallagher, A. G. Dempster, and G. Retscher, "A new method to generate and maintain a WiFi fingerprinting database automatically by using RFID," in *Proceedings of the 3rd IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1-6.
 - [48] K. Kaemarungsi, "Design of indoor positioning systems based on location fingerprinting technique," School of Information Science, University of Pittsburgh, 2005.
 - [49] Y. Zhao, Y. Liu, and L. M. Ni, "VIRE: Active RFID-based localization

- using virtual reference elimination," in *Proceedings of IEEE International Conference on Parallel Processing (ICPP)*, 2007, pp. 56-56.
- [50] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825-845, 2005.
 - [51] C.-L. Wu, L.-C. Fu, and F.-L. Lian, "WLAN location determination in e-home via support vector classification," in *Proceedings of IEEE International Conference on Networking, Sensing and Control*, 2004, vol. 2, pp. 1026-1031.
 - [52] U. Ahmad, A. V. Gavrilov, S. Lee, and Y.-K. Lee, "A modular classification model for received signal strength based location systems," *Neurocomputing*, vol. 71, no. 13, pp. 2657-2669, 2008.
 - [53] M. Moreno-Cano, M. A. Zamora-Izquierdo, J. Santa, and A. F. Skarmeta, "An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings," *Neurocomputing*, vol. 122, pp. 116-125, 2013.
 - [54] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of IEEE International Joint Conference on Neural Networks*, 2004, vol. 2, pp. 985-990.
 - [55] W. Xiao, P. Liu, W.-S. Soh, and G.-B. Huang, "Large scale wireless indoor localization by clustering and Extreme Learning Machine," in *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, 2012, pp. 1609-1614.
 - [56] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "SELM: Semi-supervised ELM with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566-2572, 2011.
 - [57] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie, "A Robust Indoor Positioning System Based on the Procrustes Analysis and Weighted Extreme Learning Machine," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252-1266, 2016.
 - [58] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155-164, 2002.
 - [59] P. Castro, P. Chiu, T. Kremenek, and R. Muntz, "A probabilistic room location service for wireless networked environments," in *Proceedings of International Conference on Ubiquitous Computing*, 2001, pp. 18-34.
 - [60] K. Weekly, H. Zou, L. Xie, Q.-S. Jia, and A. M. Bayen, "Indoor occupant positioning system using active RFID deployment and particle filters," in *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2014, pp. 35-42.
 - [61] A. Bekkali, H. Sanson, and M. Matsumoto, "RFID indoor positioning based on probabilistic RFID map and Kalman filtering," in *Proceedings*

- of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007), 2007, pp. 21-21.
- [62] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceedings of Robotics Science and Systems*, 2006.
 - [63] J. Lichtenauer, M. Reinders, and E. Hendriks, "Influence of the observation likelihood function on particle filtering performance in tracking applications," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 767-772.
 - [64] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: Location estimation using sparse and noisy sensor data," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, vol. 1, pp. 723-728.
 - [65] M. Berna, B. Sellner, B. Lisien, S. Thrun, G. J. Gordon, and F. Pfenning, "A Learning Algorithm for Localizing People Based on Wireless Signal Strength that Uses Labeled and Unlabeled Data," in *Proceedings of the 18th International Joint Conferences on Artificial Intelligence (IJCAI)*, 2003, pp. 1427-1428.
 - [66] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283-298, 2008.
 - [67] J. J. Pan, Q. Yang, and S. J. Pan, "Online co-localization in indoor wireless networks by dimension reduction," in *Proceedings of the 22th AAAI National Conference on Artificial Intelligence*, 2007, vol. 22, pp. 1102-1107.
 - [68] X. Chai and Q. Yang, "Reducing the calibration effort for probabilistic indoor location estimation," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 649-662, 2007.
 - [69] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2003, pp. 143-150.
 - [70] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, pp. 70-84.
 - [71] M. B. Kjærsgaard, "Automatic mitigation of sensor variations for signal strength based location systems," in *Proceedings of International Symposium on Location-and Context-Awareness (LoCA)*, 2006, pp. 30-47.
 - [72] M. B. Kjærsgaard and C. V. Munk, "Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength," in *Proceedings of the 6th IEEE International Conference on Pervasive*

- Computing and Communications*, 2008, pp. 110-116.
- [73] F. Dong, Y. Chen, J. Liu, Q. Ning, and S. Piao, "A Calibration-Free Localization Solution for Handling Signal Strength Variance," in *Mobile Entity Localization and Tracking in GPS-less Environments: Second International Workshop, MELT 2009, Orlando, FL, USA, September 30, 2009. Proceedings*, R. Fuller and X. D. Koutsoukos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 79-90.
 - [74] A. K. M. M. Hossain, Y. Jin, W.-S. Soh, and H. N. Van, "SSD: A robust RF location fingerprint addressing mobile devices' heterogeneity," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 65-77, 2013.
 - [75] B. J. Dil and P. J. M. Havinga, "RSS-based self-adaptive localization in dynamic environments," in *Proceedings of the 3rd IEEE International Conference on Internet of Things (IOT)*, 2012, pp. 55-62.
 - [76] Y. Zhao, N. Patwari, P. Agrawal, and M. Rabbat, "Directed by directionality: Benefiting from the gain pattern of active RFID badges," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 865-877, 2012.
 - [77] S. Yang and H. Cha, "An empirical study of antenna characteristics toward RF-based localization for IEEE 802.15. 4 sensor nodes," in *Proceedings of European Conference on Wireless Sensor Networks*, 2007, pp. 309-324.
 - [78] R. S. Parpinelli and H. S. Lopes, "A computational ecosystem for optimization: review and perspectives for future research," *Memetic Computing*, vol. 7, no. 1, pp. 29-41, 2015.
 - [79] M. Nekkaa and D. Boughaci, "A memetic algorithm with support vector machine for feature selection and classification," *Memetic Computing*, vol. 7, no. 1, pp. 59-73, 2015.
 - [80] L.-H. Chen, E. H.-K. Wu, M.-H. Jin, and G.-H. Chen, "Intelligent fusion of Wi-Fi and inertial sensor-based positioning systems for indoor pedestrian navigation," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 4034-4042, 2014.
 - [81] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715-732, 2015.
 - [82] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293-1303, 2015.
 - [83] H. E. Nyqvist, M. A. Skoglund, G. Hendeby, and F. Gustafsson, "Pose estimation using monocular vision and inertial sensors aided with ultra wide band," in *Proceedings of the 6th IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1-10.
 - [84] S. Papaioannou, H. Wen, A. Markham, and N. Trigoni, "Fusion of radio and camera sensor data for accurate indoor positioning," in *Proceedings of the 11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2014, pp. 109-117.

- [85] X. He, D. N. Aloï, and J. Li, "Probabilistic Multi-Sensor Fusion Based Indoor Positioning System on a Mobile Device," *Sensors*, vol. 15, no. 12, pp. 31464-31481, 2015.
- [86] J. R. Quinlan, "Learning with continuous classes," in *Proceedings of the 5th Australian joint conference on artificial intelligence*, 1992, vol. 92, pp. 343-348.
- [87] J. Letchner, D. Fox, and A. LaMarca, "Large-scale localization from wireless signal strength," in *Proceedings of AAAI 20th National Conference on Artificial Intelligence*, 2005, pp. 15-20.
- [88] H. Zou, L. Xie, Q.-S. Jia, and H. Wang, "An integrative weighted path loss and extreme learning machine approach to RFID based indoor positioning," in *Proceedings of the 4th IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013, pp. 1-5.
- [89] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012.
- [90] G.-B. Huang, L. Chen, and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879-892, 2006.
- [91] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1-709, 2016.
- [92] J. Torres-Sospedra *et al.*, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Proceedings of the 5th IEEE Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 261-270: IEEE.
- [93] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, and J. Huerta, UJIIndoorLoc Data Set: UCI Machine Learning Repository, 2014. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/ujiindoorloc>. Accessed on 18 June 2018.
- [94] E. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, and J. Huerta, "Wi-Fi Crowdsourced Fingerprinting Dataset for Indoor Positioning," *Data*, vol. 2, no. 4, p. 32, 2017.
- [95] E. S. Lohan, J. Torres-Sospedra, P. Richter, H. Leppäkoski, J. Huerta, and A. Cramariuc, Crowdsourced WiFi database and benchmark software for indoor positioning: Zenodo, 2017. [Online]. Available: <http://doi.org/10.5281/zenodo.889798>. Accessed on 18 June 2018.
- [96] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624-635, 2018.
- [97] M. Mohammadi and A. Al-Fuqaha, BLE RSSI Dataset for Indoor localization and Navigation Data Set: UCI Machine Learning Repository, 2018. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/BLE+RSSI+Dataset+for+Indoor+>

- [localization+and+Navigation#](#). Accessed on 18 June 2018.
- [98] G. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. Lohan, and J. Huerta, "Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning," *Data*, vol. 3, no. 1, p. 3, 2018.
 - [99] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta, Long-Term Wi-Fi fingerprinting dataset and supporting material: Zenodo, 2018. [Online]. Available: <http://doi.org/10.5281/zenodo.1066041>. Accessed on 18 June 2018.
 - [100] R. Parasuraman, S. Caccamo, F. Baberg, and P. Ogren, CRAWDAD dataset kth/rss: CRAWDAD, 2016. [Online]. Available: <https://doi.org/10.15783/C7088F>. Accessed on 18 June 2018.
 - [101] Z. Tã³th and J. Tamã;s, Hybrid Indoor Positioning Dataset from WiFi RSSI, Bluetooth and magnetometer Data Set: UCI Machine Learning Repository, 2016. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Hybrid+Indoor+Positioning+Data+set+from+WiFi+RSSI%2C+Bluetooth+and+magnetometer>. Accessed on 18 June 2018.