RADIO FINGERPRINT DATABASE ENHANCEMENT FOR INDOOR LOCALIZATION VIA CLASSICAL AND DEEP GENERATIVE MODELS

DWI JOKO SUROSO

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF ENGINEERING

IN ROBOTICS AND COMPUTATIONAL INTELLIGENCE SYSTEM

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2023

KMITL-2023-EN-D-408-031

Thesis	Radio Fingerprint Database Enhancement for Indoor Localization via Classical and Deep Generative Models
Student	Mr. Dwi Joko Suroso
Student ID.	64601003
Degree	Doctor of Engineering
Program	Robotics and Computational Intelligence Systems
Year	2023
Thesis Advisor	Prof. Dr. Pitikhate Sooraksa

ABSTRACT

Fingerprint-based Indoor localization has been known for its high accuracy performance compared to other techniques, e.g., trilateration and triangulation. However, it poses great pain points in offline database construction, including laborious effort and cost. Some proposals tried to alleviate these issues by cutting down extensive databases, i.e., crowdsourcing and database synthesis. Both methods are proven to enhance the fingerprint database but must be reduced in complexity and lightweight implementation cost. This dissertation proposes more in the database synthesis method via classical and deep generative models.

This dissertation explores the low-cost, wide-availability, and straightforward implementation of received signal strength indicator (RSSI) as a localization parameter. The classical models' radio fingerprint database enhancement using multi-features (dual features) and linear interpolation and regression techniques are proposed. For the deep generative models approaches, the variational autoencoders (VAEs) and generative adversarial networks (GANs) are applied to generate "fake" or synthetic data.

From the results, the radio fingerprint database enhancement via classical and deep generative models can help tackle offline database construction and data scarcity issues. The methods can alleviate the burden in offline fingerprint database construction by applying small or sparse databasing and compensate with the synthetic database. Moreover, radio fingerprint database enhancement can significantly improve indoor localization performance.

Acknowledgment

June 30th, 2021, at Yogyakarta International Airport, just around five minutes after my flight's check-in was open, my oldest son, Yuta, asked me to hug him. He started to cry and did not want to let me go. I kissed him, and I kissed my wife and Yuta's little brother, Seiya. We all started to cry. Loudly. Yuta did not want me to let him go from my arm. He said, you cannot go, do not leave me. He said several times. I continued crying deeply. I let my father hug Yuta. He shouted and cried. My father and my mother also cried.

Prayer and good wishes have been spoken to me from them. I cannot look back. I keep walking forward. The moment before I stepped in the entrance, I looked at everyone; I looked at Yuta, and he was still crying loudly. I waved my hand. In my heart, I said, "I will finish this Ph.D., I will work hard, I promise." *06:30 am to 06:30 pm*, every day for almost two years, proves this promise and my determination.

In KMITL, I am grateful to Prof. Pitikhate Sooraksa and Asst. Prof. Panarat Cherntanomwong inspires me to always work with my heart. "Do not worry, if you do the best as you always do, you will be fine," Prof. Pitikhate tells me almost anything. "Do not forget to take care of your body and have fun going on a trip somewhere," Prof. Panarat always reminds us to also have fun in what we are pursuing. Moreover, Farid always says, "wow! it is interesting," whenever we discuss, in or out of our research bubble. They are the persons whom I always look to for my learning and improvement. So, thank you is never enough.

Above all, I am here today, always because of a blessing from Almighty God, Allah SWT. Alhamdulillah, all praises for Allah SWT. My mother and my father, My Wife Mei, and My two sons: Yuta and Seiya, there is always tremendous and unlimited support and gratefulness. Once again, thank you words cannot describe my feeling. All Friends in KMITL; The Genk Kanyarat (Hasdar, Irfan, Thoriq, Agus, Dewi, Alim), Ari, Resty, Rita, Irfan Nurhidayat, Dio, Earth, Damian, all Indonesian friends in PERMITHA KMITL and badminton gang, the aunties from Idris canteen, whom the food always be my breakfast and lunches in the past two years, thank you for a joyful time during my study and with you guys all maybe (little) bad times always become good times, thanks for the reminder and let keep connected. Dr. Nachanant, my best friend since my master's degree here also in KMITL,

thank you for the friendship in which I will always cheer. Prof. Dr. Kosin Chamnongthai, Dr. Veerapool Monyakul, and Dr. Poom Konghuayrob, as this dissertation examiner, thank you for your comments and inputs for making this dissertation better. For lecturers, academic staffs, and students at School of Engineering, especially at Robotics and Computational Intelligence System, thank you for making my study life in KMITL enjoyable.

I thank KMITL Doctoral Scholarship for providing me with the scholarship to study and do my Ph.D. research. Thank you also for the opportunity and permission to study abroad to Dept. of Nuclear Engineering and Engineering Physics (DNEEP/DTNTF), Universitas Gadjah Mada (UGM), Indonesia and high appreciation for UGM for assisting me in the financial process for the settlement in Latkrabang as it was a really tough condition in COVID-19 pandemic time. Of course, to all the beautiful people both from KMITL and UGM: Ms. Palika, Dr. Sentagi, Dr. Alex, and others whom I cannot mention all, my thanks are not sufficient for everything they have done for me.

"Remember, only the disciplined ones are free in life," quote from Eliud Kipchoge, "Rest at the end, not in the middle," quoted by Kobe Bryan's English teacher, "On the road to achieving your dreams, you must apply discipline. But importantly, consistency. Because without commitment, you will never start, but without consistency, you will never finish," by Denzel Washington, those phrases struck me hard. As I recall, I was doing my first journal with terrible backache from 2 days working nonstop in a non-comfy chair dorm. I was 39 degrees when presenting my Doctoral Seminar; I was bleeding badly in my teeth finishing my simulation and analysis for another journal manuscript. I had terrible left ear scars inside, could not balance, and had a huge, uncomfortable feeling, but I was still working anyway. I want to discipline myself, no excuse. I was given very kind supervisors and friends and a chance for a scholarship, and I live far away from my family in Indonesia to study here at KMITL. I will never take them for granted. Once again, thank you.

Hopefully, this dissertation will be valuable and helpful for anyone who reads it, as I get many benefits from it. As a person, I am humbled and apologize for any mistakes found in this dissertation. Terima kasih.

Latkrabang, 14 March 2023

Dwi Joko Suroso

“For indeed, with hardship [will be] ease. Indeed, with hardship [will be] ease.”

~ Holy Qur’an (Al-Inshirah verse 6-7).

Just do it little by little. Everything complex will become easy.

~ My Mom

Table of Contents

List of Tables

List of Figures

Abbreviation

RSSI : Received Signal Strength Indicator

UWB : Ultra-wideband

RFID : Radio-frequency Identification

RF : Radio Frequency

CSI : Channel State Information

k-NN : k-Nearest Neighbor

NN : Neural Network

DNN : Deep Neural Network

VAEs : Variational Autoencoders

GANs : Generative Adversarial Networks

LoS : Line-of-sight

NLoS : Non-line-of-sight

WSN : Wireless Sensor Network

Chapter 1
Introduction

1.1 Background

Indoor localization is a process to determine the location or position of persons, objects, or devices inside an indoor environment. This location determination can be achieved using radio frequency (RF) or non-RF technologies [1], [2]. The most commonly used ones are RF-based technologies, e.g., Wi-Fi [3], Bluetooth[4], radio-frequency identification (RFID)[5], ultra-wideband (UWB)[6], ZigBee[7], and Lora WAN[8]. Other technologies, including vision-based, sound, and ultrasound-based systems, which less popular due to limited availability and cost for implementation [2], [9]–[11]. One of the main driving forces of indoor localization research is the unreliability of outdoor localization systems for locating objects indoors, i.e., the global navigation satellite system (GNSS), which has the global positioning system (GPS) as the well-established technology [12]. Unfortunately, GPS cannot be used reliably well in the indoor environment due to satellite signal blockage, the complex nature of multipath inside the indoor environment, and non-line of sight (NLoS) conditions[13].

The realization of the indoor localization system will benefit some services that can be enabled and its wide applications where the location information is essential. For example, indoor localization can enable the indoor Location-based service (ILBS). The Internet of Things (IoT) needs the precise location of the "things" to enable connectivity in a particular coverage. Other services, including Ambient Assisted Living (AAL), primarily assist the elderly or people needing to be located and monitored indoors, e.g., in hospitals or nursing homes. For enabling service in the emergency response, navigation is also essential in human life involvement, for instance, in the navigation of firefighters in a flamed building caused by fire. Intelligent offices or smart buildings are also enabled by indoor localization, especially to maintain energy consumption efficiency [2], [14]–[17].

Wide indoor localization applications can range from hospitality, healthcare, autonomous vehicle, shopping mall, disaster management, smart building, space exploration, sports, manufacturing, robotics, surveillance, and search. These applications are also strongly related to our daily life and use. As the service enables and applications are promptly needed, indoor localization research is part of the continuous research, especially in the wireless communication field [2], [14].

In addition to technologies, the techniques are essential for indoor localization implementation. Two main techniques are generally employed for this type of indoor localization: the range or range/distance-based and range-free (scene analysis). The range-based applies the parameter-to-distance translation, e.g., received signal strength indicator (RSSI) to the distance by path loss model, the angular properties to distance extracted by multi-antenna array system, signal flight time to distance by ranging theories. The main issue with the range-based is that the accuracy of the localization parameter needs to be high and reliable. The indoor environment has complex features with multipath effects for those parameters, so it will likely give inaccurate distance translation results. Despite that, this technique is straightforward, and in a real-life scenario, it gives fast and direct implementation [17], [18].

On the other hand, range-free, i.e., fingerprint technique, is slow. The fingerprint technique is used to locate the object or target based on similar characteristics of the wireless signal it receives. The fingerprint technique, also known as the radio map, involves offline and online phases. In the offline phase or creating "fingerprint database" phase, the fingerprints locations in the area of interest and their corresponding features/localization parameters need to be constructed." For instance, the "fingerprint nodes" measure the RSSI or other characteristics of the wireless signal from various wireless access points or "reference nodes," e.g., Wi-Fi or Bluetooth devices. These data are used to create a unique fingerprint for each location [3], [14], [15], [19], [20].

In the online or localization phase, the target will act; similarly, it receives the wireless signal parameter, then by comparing these parameters to those stored in the offline database, the localization of the target can be estimated. Typically, the pattern-matching algorithm matches the target and the fingerprint database. It calculates how similar (primarily using the minimum error) to assign the target's location. This algorithm can be represented by simple Euclidean distance or a more complex model of the machine or deep learning (ML/DL) [14], [17], [21].

The drawback of the translation issue of range/distance-based can be tackled by storing it as a fingerprint database. However, it poses the challenge of how we construct the fingerprint database in the offline phase. It is a very high cost and long effort of measurement campaign, especially when the requirement of indoor localization system is high. Furthermore, involving the workforce and inflexibility in the layout change are becoming drawbacks of the fingerprint technique. This issue is becoming the dissertation's main topic, and we specify our approach as the "Radio Fingerprint Database Enhancement [18], [22]–[24]."

This chapter will introduce the dissertation's topic, the dissertation's AOQ (aims, objectives, and questions), the significance and limitations of the research, and give the dissertation's outline and structure to navigate the flow.

1.2 Radio Fingerprint Database Enhancement

This dissertation focuses on using RF technologies and the fingerprint technique. As briefly discussed, fingerprint-based indoor localization has advantages over range-based, e.g., trilateration and triangulation, on its straightforward use of the localization parameters, e.g., RSSI, channel state information (CSI), without considering translating those parameters to the range or distances. As the definition, the fingerprint technique is a spatial location technique that uses features dependent on locations designed for localization procedures. These features are the chosen localization parameter, i.e., RSSI, known for its wide availability in RF technologies, e.g., Wi-Fi, ZigBee, Bluetooth Low Energy (BLE), and RFID. Moreover, RSSI also does not need additional hardware installation to extract/obtain the values. While CSI is famously known as a superior parameter to RSSI, its difficulty and complexity to obtain the values, yet expensive and require extensive additional hardware and software installation, hindering some actual and real-time indoor localization implementation [17], [18].

RSSI has a high sensitivity to the environment resulting from its high fluctuation values over time, so the fingerprint technique utilization becomes a suitable choice. The fingerprint technique considers all the multipath effects in the room as is, represented as the RSSI values for its corresponding fingerprint location. Fingerprint locations are design-depended on the indoor environment where indoor localization will be implemented as an area of interest [7], [25].

The fingerprint technique has two main phases: offline and online phase. The fingerprint database construction is in the offline phase, while the localization process is in the online phase. In the offline phase, the features-location dependent is carefully chosen. For example, imagine we have a typical classroom or office of a specific size; we must consider carefully how many fingerprint locations and their corresponding features, which will later form the fingerprint database.

More than sparse fingerprint locations will affect the system performance metrics, i.e., accuracy, because it depends solely on the density of these fingerprint locations. However, being too dense will affect several factors, including the high cost, adding complexity, inefficient time consumption, extensive laborious efforts, and some issues with the hardware and software size in the database construction. Moreover, the fingerprint technique must be more flexible to handle the topology change. Once we do a measurement campaign to collect the fingerprint database, changes in the layout, additional furniture, some obstructive objects, or radical change in the material of buildings will affect the fingerprint database.

The main pain point in the fingerprint technique is mainly in the database construction in the offline phase, which resulted from the accuracy requirements versus the database density consideration. Some previous publications have proposed alleviating these issues in databasing, especially the crowdsourcing method and database synthesis. Unfortunately, the crowdsourcing method needs massive effort, including many participants in data collection, which is impractical. Moreover, whether this data collection results were successful depends on the skills and attention of the people collecting the database. It also still requires a vast amount of time and high cost. In this dissertation, we focus on the latter part to explore some proposals involving the classical and ML/DL approaches for fingerprint database enhancement. For the first step, we are also eager to apply multi-feature to enhance the fingerprint database.

As we consider RF-based indoor localization, the database synthesis usually starts with the signal itself, the propagation phenomena, and its interaction with the indoor environment, i.e., the path loss model inside the room. We are continuing with some approaches to interpolation and extrapolation, using classical or sophisticated methods, i.e., ML/DL implementation. The general log-loss model is mainly used for the power-based parameter in the indoor localization system, i.e., RSSI, as it aligns with the theoretical power vs. distance calculation. We can apply interpolation and extrapolation based on

this path loss model. We can predict the power values in a specific area of interest (the indoor environment we consider in implementing an indoor localization system). Some earlier publications consider this method, and due to the complex environment, the cost of the measurement campaign (i.e., involving channel sounders) has hindered the path loss correction to the optimum results. Instead of using interpolation of power based on the log-loss model, we propose the interpolation using simple classical methods which predict the RSSI values between fingerprint locations and make the database denser.

Database synthesis via ML/DL is a growing research topic of their models' ability to recognize patterns and do complex feature extraction for helpful information. Some proposals used the idea of data synthesis or data augmentation mainly by implementing generative models. The "deep" generative models are less popular than the discriminative models. Differ from the discriminative model, e.g., multilayer perceptron (MLP), the sliding windows or famously known as the convolutional neural networks (CNNs), recurrent neural networks (RNNs), the generative model has one specific pain point or difficulties in estimating the intractable of the probability distribution, e.g., in the latent-variables model, i.e., variational autoencoders (VAEs). Ian Goodfellow et al. established generative adversarial networks (GANs) in 2014. GANs are designed by dueling (adversarial) two networks: the Discriminator and the Generator. To be able to generate new samples, these two networks have to participate in the minimax game until Nash equilibrium is achieved. This condition is established when the fake data produced by the Generator is labeled as actual data by the Discriminator.

This dissertation will discuss the fingerprint database synthesis using both explicit and implicit objective functions: VAEs and GANs. Some previous publications using RSSI and CSI have been published with their advantages and disadvantages. CSI has more information on the channel state as it has power, amplitude, and signal phase, which is essential in extracting parameters. In indoor localization, those signal properties are significant for system performance. However, it comes the price of the difficulty and cost of obtaining the CSI from standard devices, i.e., Wi-Fi. We need additional hardware, not straightforward software development, for additional disadvantages.

On the other hand, RSSI is much simpler and cost-effective, which also comes with low performance due to signal fluctuation over time. Somehow, signal strength is easily affected by environmental and physical changes. Some proposals have been widespread, including using VAEs and GANs by utilizing the RSSI. The proposals also tackle

the RSSI drawbacks of using it as a fingerprint database parameter. The deep generative model generates "fake" data or data synthesis.

1.3 Research Aims, Objectives, and Questions

As thoroughly discussed, the issue of the fingerprint database needs to solve. This research proposes radio fingerprint database enhancement for solving offline database construction burden and database sparsity and improving the system performance using a relatively small actual database.

We formulated our objectives as follows:

1. To introduce more uniqueness to the fingerprint database, implement multi-features/parameters on the fingerprint database.
2. To implement classical (interpolation and regression) and deep generative models to create a synthetic database to answer data construction burden and data sparsity.
3. To improve the localization performance by mentioned methods.

Our objectives answer some of the research questions that lay the foundation of our research topics:

1. How do we add uniqueness to the fingerprint database?
2. What methods can tackle the database construction burden and data sparsity?
3. How to improve the performance of fingerprint-based indoor localization, especially with a limited set of the database?

1.4 Research Significance and Limitations

This research benefits the fingerprint-based technique development, especially when small/sparse actual database collection is prioritized. The strategy to get more data and a denser database from this relatively small database can be applied to this dissertation's research approaches. Another benefit is that using simple, classical, or even deep learning-based database synthetic can be justified based on the environment and system's requirements.

However, there is no such perfect research; we have some limitations in our research approach. These limitations are:

1. The scope of our research is relatively narrow, which is the fingerprint technique for indoor localization for a single target.
2. Our methodology is limited in the simulation world, which still needs to be implemented in real-life positioning. We have actual data from the measurement campaign; we used those data, implemented some simulation methods, and observed the system performance.
3. With limited resources and datasets, we use several data from our previous fingerprint database attempt, which emphasized using low-cost and straightforward configuration/setup.
4. We also need more generalizability analysis since our proposed method is validated only from our dataset.

.

1.5 Dissertation Outline and Structure

This dissertation comprises 7 (seven) chapters. We have presented the "scene" of our dissertation, including what and why our research is about, in Chapter 1. Chapter 2 presents the literature review related to our research questions and lays the foundation of our research, primarily in methods we used for the objective's accomplishment. As we designed this dissertation as a "Dissertation by Publications" or "Papers Model Dissertation," we started Chapter 3 with our first journal on a multi-feature fingerprint database. We considered using the additional feature for database enhancement, illuminance values used together with the RSSI. We implemented our proposal in device-free indoor localization by the Random Forest, one of the ML algorithms for the localization algorithm.

Chapter 4 presents the fingerprint database enhancement by applying classical interpolation and regression techniques. We considered two different environments, representing 2D and 3D indoor localization. Chapters 5 and 6 apply the deep learning model, i.e., the deep generative model of explicit and implicit objective functions. Chapters 5 and 6 include our RSSI synthesis publications via deep generative models by VAEs and GANs, respectively. In addition, Chapter 6 proposes database synthesis for a relatively small measurement database and proceeds with the target position

classification via multilayer perceptron (MLP). Chapter 7 closes this dissertation by discussing and interpreting the findings of Chapters 3 – 6. Conclusions will show the full circle of this dissertation by highlighting our key finding and their implications for fingerprint database enhancement in particular and indoor localization research and application in general. Figure 1.1 shows this dissertation's structure and outline.

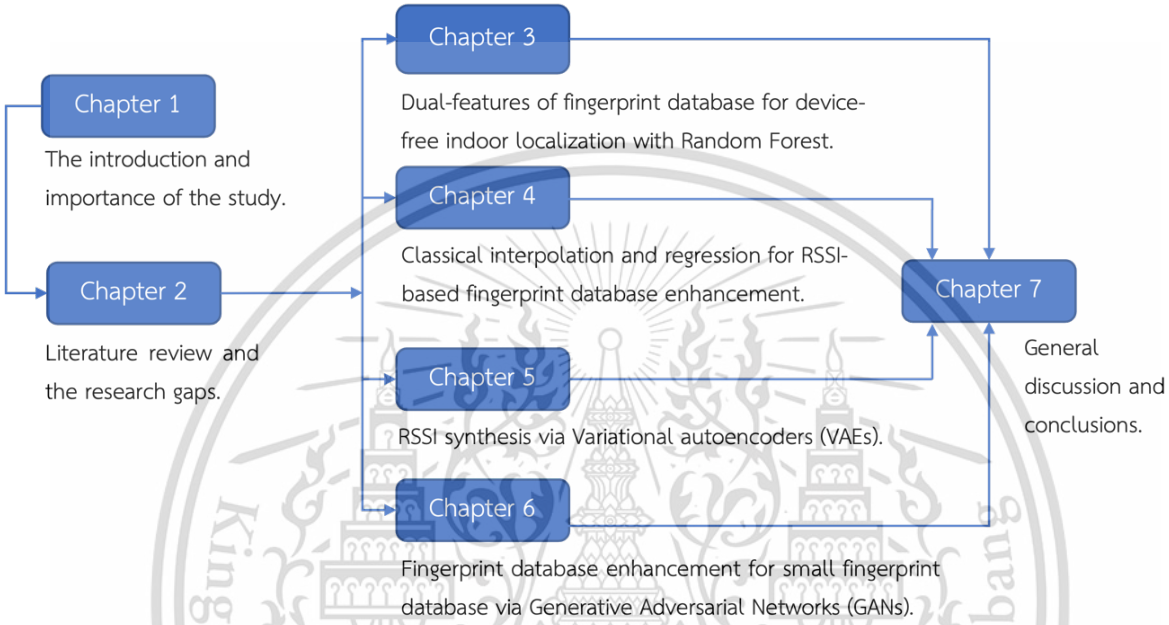


Figure 1.1 Dissertation Structure.

Chapter 2
Literature Review

Radio Fingerprint database enhancement helps tackle the issues in fingerprint-based technique database construction. This chapter will review the fingerprint technique deployment challenges and radio fingerprint database enhancement for classical and deep learning-based models. The research gaps will also be presented in the last part of this chapter. This chapter also builds the contextual framework of our proposal.

2.1 Radio Fingerprint Deployment Challenges

In the first chapter, we discussed the definition of indoor localization, its importance, and its challenges in general. This sub-chapter will discuss the challenges faced by fingerprint-based indoor localization implementation. Figure 2.1 shows the indoor localization deployment challenges. The fingerprint-based technique has challenges, including time-consuming and heterogeneous localization sources.

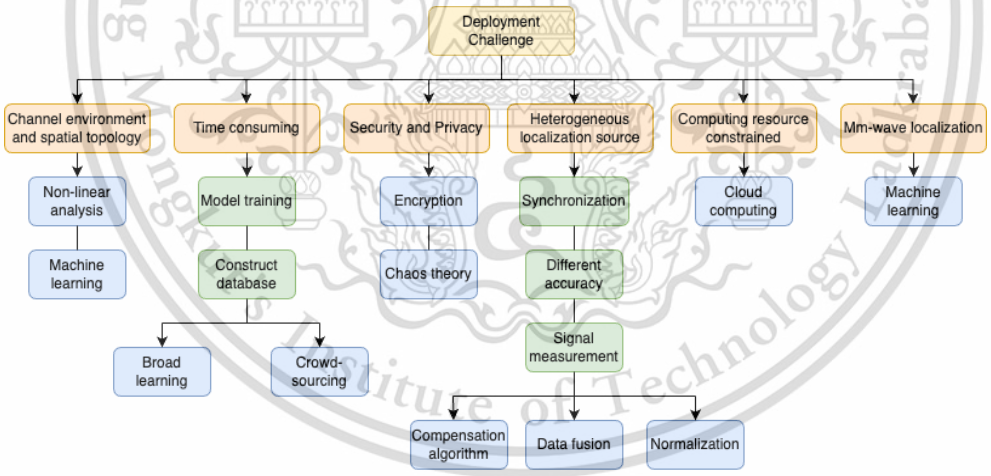


Figure 2.1 Indoor localization deployment challenges [19].

The fingerprint technique has proven satisfactory performance and is more agile in complex and non-line-of-sight (NLoS) conditions. We focus on the discussion of fingerprint technique challenges, the main issues in its database construction issue, which needs high-density grids, resulting in high cost, complexity, and time inefficiency. As our concerns, the fingerprint technique performance metrics, i.e., accuracy and precision, are heavily

dependent on the quality and density of the database. However, getting a high-density database will be more burden in the offline phase's database construction.

Moreover, the system will be more complex and labor-intensive if there is an application in a large-scale indoor environment [26]. The trade-off of this technique is that if we do not have enough density of fingerprint database, the localization performance will decrease, which yields a high localization error. On the contrary, when we build a very dense fingerprint database, multiple drawbacks will appear related to the aforementioned offline database. These challenges create opportunities for indoor localization research. How can we solve this issue? What kind of implementation of theory/algorithm? What is the research hypothesis? Our research questions, followed by aim and objectives, are presented in Chapter 1.

Cutting down extensive surveys or the database's density improvement can be one of the solutions [27]. Some proposals offered a fingerprint database based on a signal propagation model by aggregating the small number of RSSI and filling the rest using interpolation or extrapolation methods [28]–[30]. Other proposals offered a fingerprint database based on a signal propagation model by aggregating the small number of RSSI and filling the rest by using interpolation or extrapolation methods [31], [32]. Kriging interpolation is famously used for database interpolation in fingerprint-based techniques. Kriging interpolation is a spatial prediction method that uses statistical techniques to estimate unknown values based on known data points. In indoor localization, Kriging interpolation can be used to estimate the location of a target device by predicting its signal strength based on the signal strength measurements from nearby access points. Specifically, in fingerprint-based indoor localization, Kriging interpolation can be used to interpolate the signal strength values at unobserved locations based on the measured signal strengths at observed locations, which can then be used to estimate the location of the target device using the fingerprinting algorithm [33]–[35]. However, some complex issues and the accuracy results of Kriging, particularly RSSI parameter cases, are not fulfilled the expectation [36]. The other approaches use the image processing method in which the grayscale image is applied to convert the RSSI values in the area of interest [37]. The imaging method is promising but highly complex in the super-resolution image conversion from the RSSI to grayscale image conversion. The ML/DL-based method for database augmentation has abundantly been proposed. Because of the threshold in the number of data required in some ML techniques, the complexity is also high [38]–[40].

In this dissertation, we propose the radio fingerprint database enhancement via classical and deep learning-based models for relatively small databases and acquiring from low-cost devices, which could have impacted the ease of fingerprint-based indoor localization deployment. The scope and limitations of our proposal are also presented in Chapter 1. Using the most straightforward algorithm with as low computational as possible is preferable. The rest of this chapter will discuss the individual approach of classical and deep learning-based, with the closing part on the research gaps.

2.2 Radio Fingerprint Database Enhancement via Classical Method

The classical method approach here is more to the simple calculation method for predicting the value of the similar localization parameter in the context of their fingerprint location. Also, it means using dual or more localization parameters straightforwardly (without sensor fusion in this case) for the fingerprint database. Sensor fusion has different complexity and high computation load. We will start by discussing the dual parameter for the fingerprint technique. Uniquely, we applied the dual parameters for the Indoor device-free localization (IDFL), which is less popular than the conventional, i.e., device-based indoor localization system.

Moreover, we apply Random Forest, a machine learning algorithm, to assist in the pattern-matching algorithm for determining the target's location. The last subsection uses interpolation and regression for database enhancement in 2D and 3D environments. The goal of this classical approach is simplicity yet acceptable even the high performance.

2.2.1 Fingerprint Database Enhancement via Dual or Multi-Features

Youssef et al. introduced the IDFL, meaning that an object or a person can be localized without any device attached [41]. This concept provides more flexibility but is challenging regarding the parameter used for localization. For example, in the device-based, the parameter, i.e., distance, can be used from the target to neighboring references, taking the geometrical approach to estimate the target's location. Furthermore, the IDFL relies heavily on the parameter differences between existing and does not exist a person or the target in an indoor environment. The commonly used technology for IDFL is computer vision-based technologies, e.g., RGB camera, depth camera, and infrared sensors [42]–[44]. However, they have privacy-obstruction issues. Unlike the device-based, the primary technique used in IDFL is exclusively only

fingerprinting-based and radio tomography imaging (RTI), as the authors' concern. The most used technologies for IDFL are radio-based (privacy-issue-free), predominantly by Wi-Fi, ZigBee, RFID, and UWB-Radar [11].

The main difference between IDFL and device-based localization is that the intended target does not need to bring receiving devices [45]. The first development or early fingerprint-based application RSSI was because of its low-cost and straightforward implementation on several available RF devices and known as "passive" localization [41], [45], [46]. Another commonly used localization parameter is CSI, which can give more information on propagation channel properties in the area of interest [25], [47]. However, using CSI is not straightforward and needs some additional procedures and installations in the current RF devices [48]. These reasons are behind the high cost and complexity of its implementation for indoor localization, especially IDFL.

In this context, we focus on discussing the classical way to use dual or more localization parameters used in fingerprint-based IDFL systems, both sensor and radio-based, especially the use of RSSI as its benefits have been discussed aforementioned. We summarize the literature for fingerprint database enhancement by applying this dual or multi-feature/parameter approach. In this sub-chapter, we will review based on the use of low-cost device implementation, mainly Wi-Fi, with RSSI as the core parameter and another additional feature(s) from sensor technologies. In addition, we also dive deep into machine/deep learning implementation, helping the IDFL system to be more reliable.

The most common available literature uses a dual or multi-parameter, i.e., RSSI with CSI [49], [50]. These published papers proposed a hybrid fingerprint location technology based on RSS and CSI to improve indoor location accuracy and algorithm efficiency—the combination of these parameters usually for tackling the instability and low space distinguishability of traditional RSSI-based systems. However, as we are concerned, dual feature use is still uncommon, e.g., RSSI with the other sensor's parameters, e.g., illuminance and inertial measuring unit (IMU). One research group discussed RSSI from Wi-Fi combined with the visual features from the scene, e.g., EXIT signs are unique to a specific position as Wi-Vi fingerprint [51], [52]. Another publication proposed localization system based on the fusion of RSSI and magnetometer sensor data. The sensor fusion procedure is complex and has a degree of complexity. The system used an ML algorithm, Random Forest, and Extreme Gradient Boosting (XG-

Boost) to learn the relationship between the sensor data and the target person's location. The results showed that the proposed system achieved a localization accuracy of 98% in self-collected data. The paper claimed that the simple ML algorithm could perform with high accuracy. However, this paper considered the device-based system [53]. Similar to the previous proposal, there is a publication aimed at improving room-level localization in crowded spaces using IoT systems by testing the combination of RSSI and accelerometer features through ensemble learning methods. The study found that RSSI extracted features performed better, while the individual performance of the accelerometer was poor and affected the fusion results. The classification algorithms and fusion methods were evaluated using different protocols applied to a public dataset [54]. Once again, this publication does not consider the IDFL system.

Considering IDFL and the dual/multi-feature motivates the research in [55]. The study aimed to improve device-free indoor localization by leveraging human-object interaction (HOI) events in a furnished home through a Bayesian probabilistic framework that integrates RSSI signals and HOI events, achieving an average accuracy of 95% in localizing a resident and a mean error distance of 0.58m in tracking a moving subject. The study highlights the importance of considering HOI contexts in cluttered environments to enhance the performance of commercial off-the-shelf (COTS) RFID-based localization systems. The effectiveness of the proposed method was demonstrated through experiments conducted in a residential house. The room for consideration is the selection of technology; the papers mentioned above also used RFID, as in many references discussed in Chapter 1, Wi-Fi is preferable.

Table 2.1 summarizes some published works in dual or multi-features methods. Some of the papers considered the device-based, and based on Ruan et al., the additional sensors, i.e., pervasive sensors proven to give high accuracy and small error distance [55].

Table 2.1 The dual or multi-parameter for indoor localization.

Author (year)	Additional Parameter	Evaluation Frameworks	
		Method	Results
Chandra (2022) [53]	Magnetic sensors.	Non-IDFL. Combining RSSI and magnetic sensor measurement, localization was done by applying ML algorithms (Random Forest and XGB).	By using only 5% of the original data set for interior location detection and demonstrated the potential to scale up the approach to larger infrastructures with minimal data collection and prediction time.
Tsanousa (2021) [54]	Accelerometer sensors.	Non-IDFL. Added value of using an accelerometer along with RSSI for room-level localization and assess the performance of ensemble learning methods.	Achieved better performance of the RSSI extracted features, while the accelerometer's individual performance was poor and subsequently affected the fusion results.
Ruan (2016) [55]	Pervasive sensors.	IDFL. Considering pervasive sensors for revealing people's interleaved locations during daily living activities. The RFID is selected as the core of system.	Successfully localized a resident with average 95% accuracy and track a moving subject with 0.58*m* mean error distance.

In our hypotheses, the dual features can increase the uniqueness of the fingerprint technique, in which we expect the localization performance to be improved. Most considerations in dual or multi features/parameters employed sensor fusion, which is not straightforward and highly complex. There are still few approaches in dual/multi-localization parameters for IDFL in simple yet effective results. In this dissertation, we

consider the use of an illuminance sensor together with RSSI since both parameters can be obtained at a relatively low cost and their flexibility to be used together. For instance, the RSSI can be obtained using a low-cost communication device (Wi-Fi-based). By attaching the illuminance sensor to the same devices, the sensor data can be sent together with RSSI for the exact fingerprint locations. We also employ the Random Forest as target classifier, and compare the performance to other ML models, e.g., k-NN, NN.

2.2.2 Fingerprint Database Enhancement via Interpolation Techniques

The main drawback of the fingerprint technique is its dense database construction in conjunction with the demand/requirement of localization system performance. The high-density database needs a relatively high cost and complexity of its construction in the offline phase. One method to increase the density of the fingerprint database (virtual fingerprint locations) is the interpolation method. There are several interpolation methods ranging from low to high complexity. We are motivated to apply the simple or basic interpolation method to have benefits in a less complex system. Several published literatures consider the various interpolation methods shown in Table 2.2.

Table 2.2 Related works on interpolation techniques for fingerprint-based technique.

Author (year)	Interpolation Techniques	Evaluation Frameworks		
		Param.	Method	Results
Gu, et al. (2016)[56]	SRSVD combined with k-NN	RSSI	Presented as merging matrix. Filling the missing columns and rows in the matrix.	Recovering all the fingerprint information with error rate below 6.6%. By only just 5% of the data, the approach can recover the information (error rate below 14%, without loss of localization accuracy).

Talvitie, et al. (2015) [57]	Linear, minimum, mean, gradient method extrapolation	RSSI	Interpolation and extrapolation of RSSI for 2.4 and 5GHz WLAN.	Avoiding certain error on interpolation and extrapolation RSSI is significant to give better performance results.
Moghtadaiee, et al. (2019)[58]	Zone-based PLM, zone-based WRB	RSSI	Reconstruct database on a low RPs measurement.	By using only 11% of RPs, the new zone-based PLM can reduce 26% localization error compared to conventional PLM. The zone-based WRB improve the accuracy by 40% and superior to interpolation benchmark techniques.
Ali, et al. (2017) [59]	Sparse recovery techniques.	RSSI	Introducing novel schemes for indoor localization, outlier detection, and radio map interpolation.	The method performs better compared to previous method on a small number of access points and finer granularity of RPs.
Zayets, et al. (2018) [60]	Multipath components inter/extrapolation.	CSI	Novel algorithm for CSI and multipath fingerprints (positions) as VTs.	The interpolation scheme proposed reduce 20% average localization error.

Li, et al. (2019)[61]	Local-to-local fingerprint.	RSSI	Coarse-gained fingerprint database to dense database by augmenting artificial data for radio map.	The overall average distance error is reduced. The proposed method also discusses the density requirement for adding the artificial data.
Be, at al. (2018)[62]	Inverse distance weighted (IDW).	RSSI	Crowdsourcing and interpolation methods are introduced for constructing radio map.	Computational cost is reduced by PCA. Consistent localization accuracy is achieved even only use 25% of total reference points.
Mao, et al. (2018) [35]	Kriging	TX Power	Combining residual maximum likelihood-based radio propagation parameter estimation with Kriging-based transmission power prediction.	Interpolation accuracy can be improved by 2 dB.
Zhao, et al. (2016) [34]	Kriging	RSSI	Combining the universal Kriging interpolation, k-NN and Naïve-Bayes classifier (NBC).	Achieving mean error of 1.265m, better than the traditional IPS with 28 observation points and is even comparable to the traditional IPS with 112 observation points.

Zuo, et al. (2018) [33]	Kriging	RSSI	To generate and update the database efficiently.	Proven to have practical values and poses the acceptable performance results.

*Sparsity Rank Singular Value Decomposition (SRSVD), Path loss model (PLM), Weighted Ring-based (WRB), reference points (RPs), virtual transmitters (VTs).

A published paper discussed interpolation and extrapolation methods for RSSI-based fingerprint technique [57]. The paper proposed a solution to fill the incomplete fingerprint database on WLAN, resulting in the proper interpolation and extrapolation methods that can improve the user's horizontal positioning accuracy and floor detection probability. The comparison of several interpolation techniques for RSSI-based fingerprint technique is discussed in our latest paper in [36]. The authors proposed enhancing the fingerprint database using various spatial interpolation methods, including Inverse Distance Weighted, Quadratic Spline, Cubic Spline, and Ordinary Kriging. The results show that the synthetic RSSI data generated using these interpolation methods provided a lower prediction error and had similar accuracy performance to manual fingerprints using actual data. The proposed methods can enhance the fingerprint database and improve localization performance, with a 0 dBm error for the best prediction and not more than 6 dBm for the worst prediction. The new "denser" database via IDW and ordinary Kriging gives the best localization performance.

While some publications discuss the complex interpolation techniques to some degree, is it possible yet practical to employ only simple interpolation techniques for this fingerprint database? Another consideration has also related to using the interpolation method in a 3D environment, which usually represents the multistory building. The 3D indoor localization still needs to be improved as far as we are concerned.

2.3 Radio Fingerprint Database Enhancement via Deep Generative Models

Machine learning (ML) is a type of algorithm which typically used for data analysis, computer vision (CV), pattern recognition, signal processing, natural language processing (NLP), and other which mainly have the data as the central point. Machine learning is a

subset of artificial intelligence (AI), and it has subset under it, which is deep learning (DL). Figure 2.2 illustrate the artificial intelligence set of models.

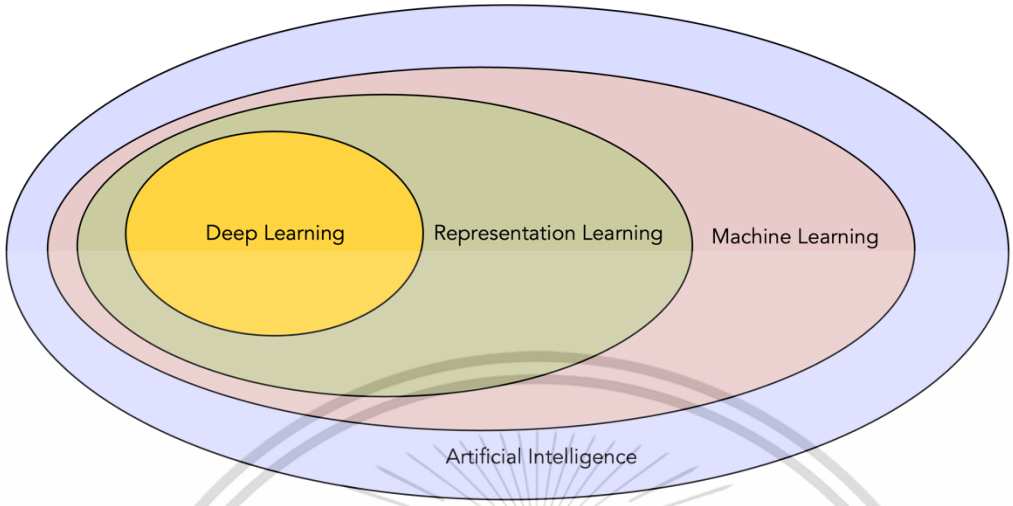


Figure 2.2 Artificial Intelligence [63].

Neural networks (NNs) are the core of DL. As the name suggests, it imitates a neural system like the human brain. Terms deep are how we can train the computer or machine to learn things "deeply." An excellent and deep-trained machine can result in high-performance accuracy. The excellent means that the fulfilled hyperparameter and other considerations help fit the model well (good training). The deep does not necessarily have many hidden layers, but it is more to an optimum number of hidden layers, which depend on the type and amount of data. DL can handle high-dimensional data and efficiently focuses on the right features. This process is called automatic feature extraction. The fundamental difference between ML and DL can be illustrated in Figure 2.3.

The fundamental difference is that the manual features (attributes) selection is not needed in DL. Also, it calls "deep" for the n the number of hidden layers compared to ML, which utilizes more shallow or less hidden layers. Feature selection in ML can be a collection of images on the cat face anatomies, i.e., eyes, nose, and ears, while DL needs the set of cat face images and extract features by its "deep" neural networks. As in the previous example, DL has a crucial idea to learn from raw data. DL is now very popular in almost all research fields because of the advanced development of hardware, i.e., graphical processing unit (GPU), software development free or by subscription, including the language program, and big data. DL generally is comprised of two models: the discriminative and generative models. The discriminative model acts as a classifier or regressor in the ML model. In contrast, the generative model idea is to produce new data

similar to the input data used for learning. Our research focuses on using a deep generative model (DGM) for RSSI synthesis.

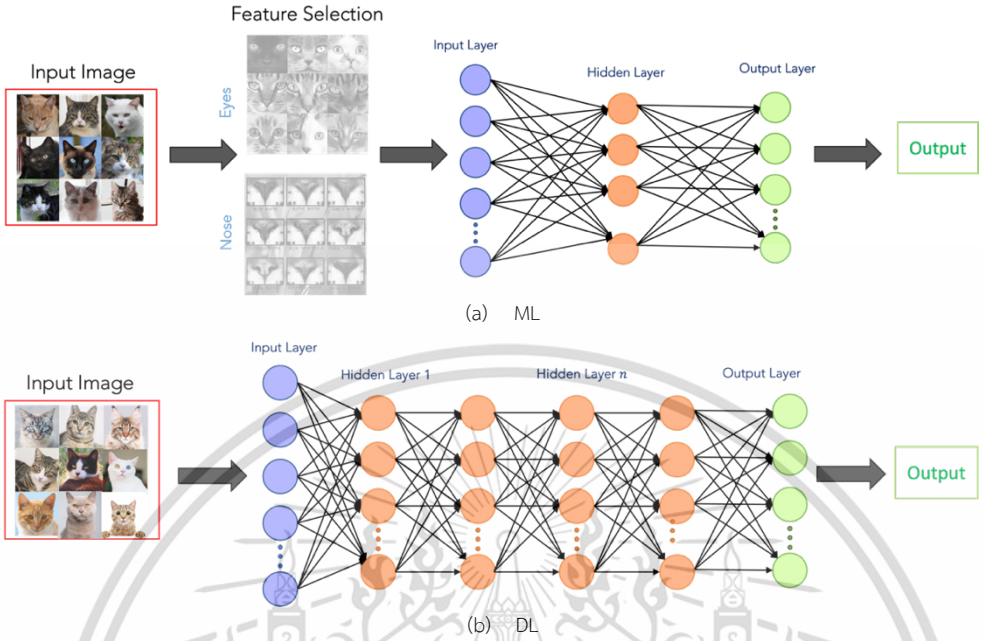


Figure 2.3 ML vs DL fundamental different.

2.3.1 Deep Discriminative Models

Deep neural networks (DNNs) are the most form of deep discriminative models. One of the most applied and well-established DNNs is convolutional neural networks (CNNs). A deep discriminative model predicts the unseen data based on the conditional probability that can be used for classification or regression purposes. This model is also known as the conditional model (learning boundaries between classes/labels in the dataset) and is considered supervised learning [63].

2.3.2 Deep Generative Models

Ian Goodfellow, the author of Generative Adversarial Networks (GANs) stated "You can think of generative models as giving artificial intelligence a form of imagination" [64]. A long history of machine or deep learning is used for classification or regression. At the same time, the generative model is not attracting interest because of several obstacles, including how to estimate the intractable dataset distribution to produce new data. Not until the recent excitement of vast application of the deep generative model in generating the image, sound, and social media filters. Several reasons to study generative models, including but not limited to [65]:.

1. The generative model can give us insight into how to represent and manipulate high-dimensional probability distributions via training and sampling. While as we know that high-dimensional probability distribution has vast applications in math and engineering.
2. The generative model can incorporate model-based reinforcement learning. It helps planning because it can learn the current state's distribution to predict the next state's distribution.
3. The generative model can be trained with missing data, and finally, it can predict the input that the data is missing. The most used is in the semi-supervised learning of the generative model.
4. The generative model can give machine learning working with not only single mode but also multi-mode or multi modals.

The generative model has recently become a crucial part of literature [66]. We also witness generative modeling tasks, i.e., single-image super-resolution, where we can produce a high-resolution image. Moreover, it can create art or maps from only basic sketches, then image-to-image, or even video-to-video translation, where we can change the video of a horse to a zebra [67]. To explain the basic concept of the deep generative model, let us see an example of maximum likelihood. Maximum likelihood has a primary idea of how a model can estimate the probability distribution. In general, this probability distribution is parameterized by the parameter $\boldsymbol{\theta}$. From here, we state the "likelihood" as the model's probability that assigns to training data $\prod_{1=1}^{m} p_{model}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta})$, m represents several training samples of the respective dataset. Then, we can observe that the main principle of maximum likelihood is to select the model's parameters that maximize the training data's likelihood [65]. Figure 2.4 shows the deep generative model taxonomy [68].

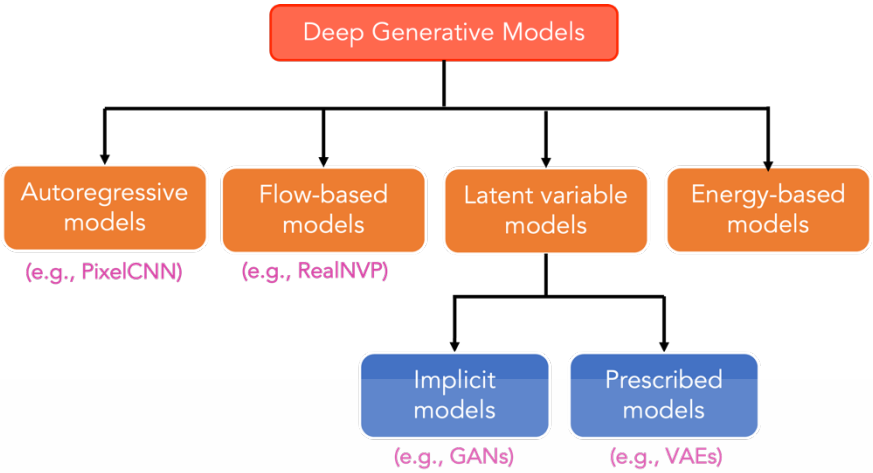


Figure 2.4 Deep generative model taxonomy.

The generative model is a relatively less considered and relatively newer area in the deep learning research environment. The generative model is somehow really challenging, including how difficult to estimate the posterior probability in the data (intractable). The basic understanding of the generative model is to learn the distribution of the input and generate new values as the output by the latent distribution learned from the input distribution. We implement the Variational Autoencoders (VAEs) and the Generative Adversarial Networks (GANs) for RSSI synthesis. Figure 2.5 depicts the AI landscape's relationships between AI, ML, and generative models for VAEs and GANs.

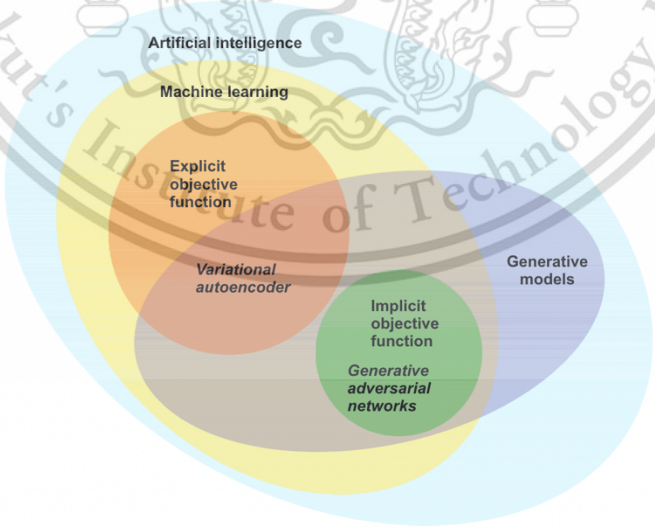


Figure 2.5 VAEs and GANs in AI's perspective [66].

2.3.2.1 Explicit Density (Prescribed) Model: Variational Autoencoders (VAEs)

We need information about the probability density function in an explicit or prescribed generative model. Let, for example, $\boldsymbol{p_{model}(x, \theta)}$, which we need to use or learn true/actual data $\boldsymbol{x}$, to fit the parameters, $\boldsymbol{\theta}$. After training, the model can produce a new example from the trained model or distribution. The explicit density model, i.e., the maximum likelihood estimation (MLE) approximates inference, e.g., VAEs and the Markov chain method [25] [63]. VAEs are the improved version of traditional autoencoders based on variational inference proposed by [69]. The notation used for the algorithm is detailed in Table 2.3 [70].

Table 2.3 Notation.

Symbol	Meaning
$\boldsymbol{\mathcal{D}}$	The dataset, $\boldsymbol{\mathcal{D}} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$, containing n data sample, $\boldsymbol{\lvert\mathcal{D}\rvert = n}$.
$\mathbf{x}^{(i)}$	Each of data point is a vector of d dimensions, $\mathbf{x}^{(1)} = \left[x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}\right]$.
$\mathbf{x}$	One data sample from dataset, $\mathbf{x} \in \boldsymbol{\mathcal{D}}$.
$\mathbf{x}'$	The reconstruction of $\mathbf{x}$.
$\tilde{\mathbf{x}}$	The noisy/corrupted version of $\mathbf{x}$.
$\mathbf{z}$	The latent (the compressed code learned in the bottleneck layer)
$\boldsymbol{a_j^{(l)}}$	The activation function for the j-th neuron in the l-th hidden layer.
$g_\phi(.)$	The encoding function parameterized by ϕ.
$f_\theta(.)$	The decoding function parameterized by θ.
$g_\phi(\mathbf{z}\lvert\mathbf{x})$	The probabilistic encoder (estimated posterior probability function).
$p_\theta(\mathbf{x}\lvert\mathbf{z})$	The probabilistic decoder (Likelihood of generating true data sample given the latent code).

Autoencoder

Before we dive deeper into VAEs, let's consider a brief autoencoder concept explanation. Autoencoder is mainly used for dimensionality reduction, and it is built by a neural network model with a narrow middle bottleneck [71]. This bottleneck layer captures a compressed latent encoding from the encoder. The illustration of autoencoder model architecture is depicted in Figure 2.6.

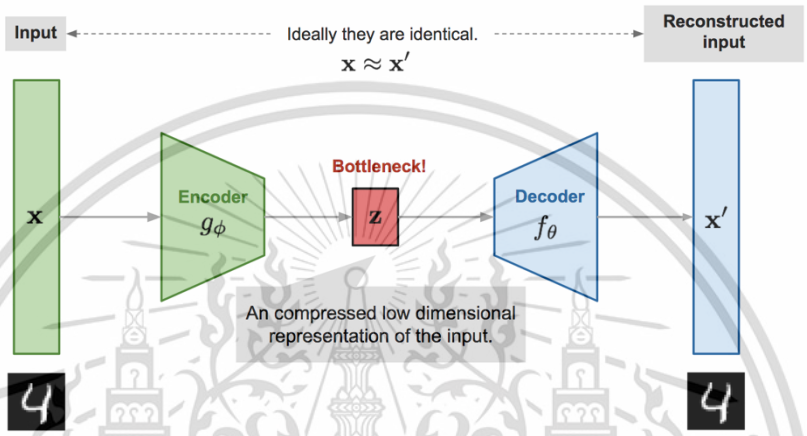


Figure 2.6 Traditional autoencoder.

Autoencoder consists of two networks: Encoder and Decoder. Encoder translated the high-dimensional input into the latent low-dimensional code (or latent, $\mathbf{z}$). While then, the decoder recovers the data from the latent with larger output layers). The encoder network works like Principal Component Analysis (PCA) to reduce dimensionality. A good intermediate representation will have benefit points: can capture the latent variable well and fully decompression later.

As shown in Figure 2.5, the workflow of autoencoder is that we have an input, $\mathbf{x}$, then, the model comprises of an encoding function of this input as $\boldsymbol{g_\phi(.)}$ and a decoding function $\boldsymbol{f_\theta(.)}$. The low-dimensional code learned for input $\mathbf{x}$ in the bottleneck layer is $\mathbf{z} = \boldsymbol{g_\phi(\mathbf{x})}$ and the reconstructed input is $\mathbf{x}' = \boldsymbol{f_\theta\left(g_\phi(\mathbf{x})\right)}$. The parameters $\boldsymbol{(\theta, \phi)}$ are learned simultaneously to produce the reconstructed data sample same as the original input (to learn identity function), $\mathbf{x} \approx \boldsymbol{f_\theta\left(g_\phi(\mathbf{x})\right)}$. The performance metric for autoencoder varies, to show the difference between vector $\mathbf{x}'$and $\mathbf{x}$, we can simply use mean square error (MSE) loss as:

$$\mathcal{L}_{AE}(\theta,\phi) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}^{(i)} - \boldsymbol{f}_{\theta}\left(\boldsymbol{g}_{\phi}\left(\mathbf{x}^{(i)}\right)\right)\right)^2 \quad (2.1)$$

Variational Autoencoders (VAEs)

VAEs is quite different from its predecessor, and it is more deeply rooted in variational Bayesian and graphical models. Variational Bayesian enables statistical inference where the value of the random variable is given from another random variable as an optimization function that finds the parameter values that minimize the objective function. The decent difference is that instead of mapping an input into a single vector (at the middle bottleneck), VAEs map the input into distribution. Let label this distribution as p_{θ}, parameterized by θ.

The relationship between original data input, $\mathbf{x}$ with the latent encoding vector, $\mathbf{z}$, is defined by three probability distributions: prior $p_{\theta}(\mathbf{z})$, likelihood $p_{\theta}(\mathbf{x}|\mathbf{z})$, and posterior $p_{\theta}(\mathbf{z}|\mathbf{x})$. Let's assume that we know the real parameter, θ^* these three distributions, the steps to generate sample output like the real data point $\mathbf{x}^{(i)}$ as:

1. Sampling a $\mathbf{z}^{(i)}$ from prior distribution $p_{\theta^*}(\mathbf{z})$
2. Value $\mathbf{x}^{(i)}$ is then generated by the conditional distribution $p_{\theta^*}\left(\mathbf{x}\middle|\mathbf{z}=\mathbf{z}^{(i)}\right)$

The parameter which maximizes the probability of generating real data samples is the optimal parameter, θ^*.

$$\theta^* = \arg\max \prod_{i=1}^{n} p_{\theta}\left(\mathbf{x}^{(i)}\right) \quad (2.2)$$

Rewrite the Eq. (3) in the log form:

$$\theta^* = \arg\max \sum_{i=1}^{n} \log p_{\theta}\left(\mathbf{x}^{(i)}\right) \quad (2.3)$$

Note: $\arg\max$ with respect to θ.

Then, updating equation and involving the laten encoding vector as,

$$p_{\theta}\left(\mathbf{x}^{(i)}\right) = \int p_{\theta}\left(\mathbf{x}^{(i)}\middle|\mathbf{z}\right) p_{\theta}(\mathbf{z})\, d\mathbf{z} \quad (2.4)$$

This is the challenge is emerged, to calculate $p_\theta(\mathbf{x}^{(i)})$ it is difficult since in the right-hand side expression, we need to check all possible value of $\boldsymbol{z}$, yet expensively, then sum them up. Here, the VAEs as graphical model offer a solution. To make faster search, the new approximation function of output is introduced as $q_\phi(\mathbf{z}|\mathbf{x})$, input of $\mathbf{x}$ and parameterized by $\boldsymbol{\phi}$.

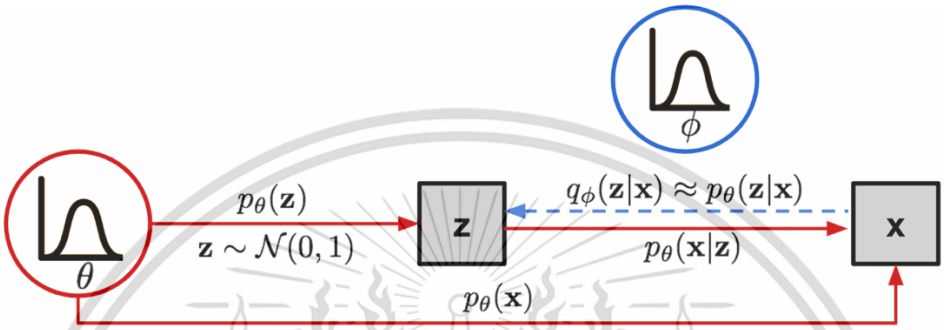


Figure 2.7 Graphical model in VAEs [70].

Figure 2.7 illustrates probability approximation in which a solid line represents the generative distribution $p_\theta(.)$. While dashed line shows the distribution $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate the intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$. From here, the process is then like an autoencoder, which:

- The conditional probability $p_\theta(\mathbf{x}|\mathbf{z})$ defines a generative model, like the decoder function $f_\theta(\mathbf{x}|\mathbf{z})$. $p_\theta(\mathbf{x}|\mathbf{z})$ is also known as the probabilistic decoder.
- The approximation function, $q_\phi(\mathbf{z}|\mathbf{x})$, is the probabilistic encoder, playing a similar role as the encoding function of autoencoder, $g_\phi(\mathbf{z}|\mathbf{x})$.

Evidence Lower Bound (ELBO)

The estimated posterior $q_\phi(\mathbf{z}|\mathbf{x})$ should be very similar to the $p_\theta(\mathbf{z}|\mathbf{x})$. To achieve that, we can apply Kullback-Leibler (KL) divergence, which estimates the similarity between two probability distributions or quantify the "distance" between estimated and actual posterior. For example, the notation of KL divergence as $D_{KL}(X|Y)$ shows how much information is lost if the distribution of Y is used to represent the distribution of X. For this case, we want to minimize the $D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z}|\mathbf{x})\right)$ with respect to parameter $\boldsymbol{\phi}$. This minimizing value

process follows the reverse KL divergence which is justified as illustrated in Figure 2.8.

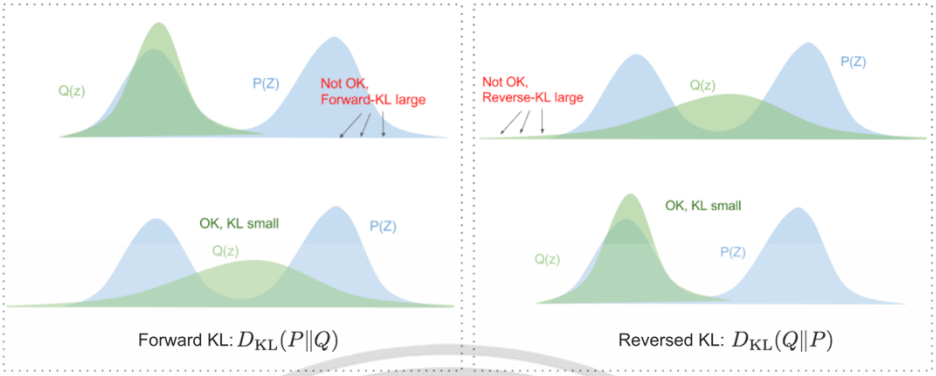


Figure 2.8 Forward and reversed KL divergence (image: blog.evjang.com/2016/08/variational-bayes.html)

- Forward KL divergence will have the expression of $D_{KL}(P|Q) = \mathbb{E}_{z\sim P_z(z)} \log \frac{P(z)}{Q(z)}$; to apply this we must make sure that $Q(z) > 0$ wherever $P(z) > 0$. As shown in Figure 2.7 (left), the optimized variational distribution $q(z)$ must cover the entire $p(z)$. Resulting that the $Q(z)$ and $P(Z)$ are different (not minimizing the distance).
- While reverse KL divergence, $D_{KL}(Q|P) = \mathbb{E}_{z\sim P_z(z)} \log \frac{Q(z)}{P(z)}$; minimizing the reverse KL divergence squeeze the $Q(z)$ under $P(z)$.

The equation can be expanded based on the reversed KL divergence as:

$$
\begin{aligned}
& D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \middle| p_\theta(\mathbf{z}|\mathbf{x})\right) \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x}) p_\theta(\mathbf{x})}{p_\theta(\mathbf{z},\mathbf{x})} d\mathbf{z} && ; p(z|x) = p(z,x)/p(x) \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \left(\log p_\theta(\mathbf{x}) + \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z},\mathbf{x})} \right) d\mathbf{z} \\
&= \log p_\theta(\mathbf{x}) + \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z},\mathbf{x})} d\mathbf{z} && ; \int q(z|x)dz = 1 \\
&= \log p_\theta(\mathbf{x}) + \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})} d\mathbf{z} && ; p(z,x) = p(z|x)p(x) \\
&= \log p_\theta(\mathbf{x}) + \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z})} \right. \\
&\qquad \left. - \log p_\theta(\mathbf{x}|\mathbf{z}) \right]
\end{aligned}
$$

$$= \log p_\theta(\mathbf{x}) + D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z})\right) - \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) \tag{2.5}$$

Thus, the Eq. (2.5) is fully described of reversed KL divergence as:

$$D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z}|\mathbf{x})\right) = \log p_\theta(\mathbf{x}) + D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z})\right) - \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) \tag{2.6}$$

Rearrange the above equation, we will have:

$$\log p_\theta(\mathbf{x}) - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z})\right) \tag{2.7}$$

The left-hand side expression of Eq. (8) is our objective to minimize when learning with the true distribution. This means that we want to maximize the (log-) likelihood of generating the real data, $\log p_\theta(\mathbf{x})$ at the same time to minimize the difference between the estimated and real posterior distribution, in this case the KL divergence acts as regularizer. Additionally, $p_\theta(\mathbf{x})$ is fixed with respect to q_ϕ. Our VAEs loss function then can be written as:

$$\begin{aligned}\mathcal{L}_{VAEs}(\theta,\phi) &= -\log p_\theta(\mathbf{x}) + D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z}|\mathbf{x})\right) \\ &= \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z})\right) \\ &= \arg\min \mathcal{L}_{VAEs}\end{aligned} \tag{2.8}$$

*$\arg\min$ with respect to θ, ϕ.

In Variational Bayesian method, $\mathcal{L}_{VAEs}$ is known as the variational lower bound or evidence lower bound (ELBO). The "lower bound" indicates that KL divergence value is always non-negative nd thus $-\mathcal{L}_{VAEs}$ is the lower bound of $\log p_\theta(\mathbf{x})$. Therefore,

$$-\mathcal{L}_{VAEs} = \log p_\theta(\mathbf{x}) - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\middle|p_\theta(\mathbf{z}|\mathbf{x})\right) \leq \log p_\theta(\mathbf{x}) \tag{2.9}$$

As the equation suggests, by minimizing the loss, we are maximizing the lower bound of the probability of generating the real data samples.

Reparameterization Tricks

The expectation form in the Eq. (2.7) invokes producing samples from the $\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})$. The nature of this sample is a stochastic process which prevents us to do backpropagation in training. In VAEs, the reparameterization trick is introduced so that the model can be trained. Let express a random variable $\mathbf{z}$ as a deterministic variable, $\mathbf{z} = \mathcal{T}_\phi(\mathbf{x}, \boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon}$ is the auxiliary independent random variable, and the $\mathcal{T}_\phi$ as a transformation function which parameterized by ϕ converts $\boldsymbol{\epsilon}$ to $\mathbf{z}$.

The common choice of the form of $q_\phi(\mathbf{z}|\mathbf{x})$ is the multi-variate Gaussian distribution with a diagonal covariance structure, as:

$$\begin{aligned} \mathbf{z}\sim q_\phi\left(\mathbf{z}|\mathbf{x}^{(i)}\right) &= \mathcal{N}\left(\mathbf{z}, \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I}\right) \\ \mathbf{z} &= \boldsymbol{\mu} + \boldsymbol{\sigma}\odot\boldsymbol{\epsilon}, \end{aligned} \tag{2.10}$$

where $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0}, \boldsymbol{I})$; reparameterization trick and $\odot$ is an element-wise product. Figure 2.9 illustrates of this reparameterization trick.

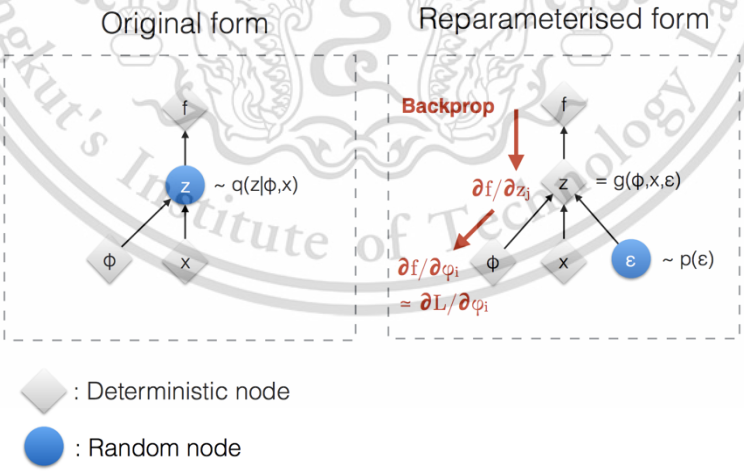


Figure 2.9 Reparameterization trick [72], [73]

From the example distribution of multivariate Gaussian, the mean and variance are learnt to make the training possible (trainable), the μ and σ, explicitly using the reparameterization trick. As note, the stochastic process remains the

same in the random variable of $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. For the multivariate Gaussian, the VAEs process can be shown in Fig. 2.10.

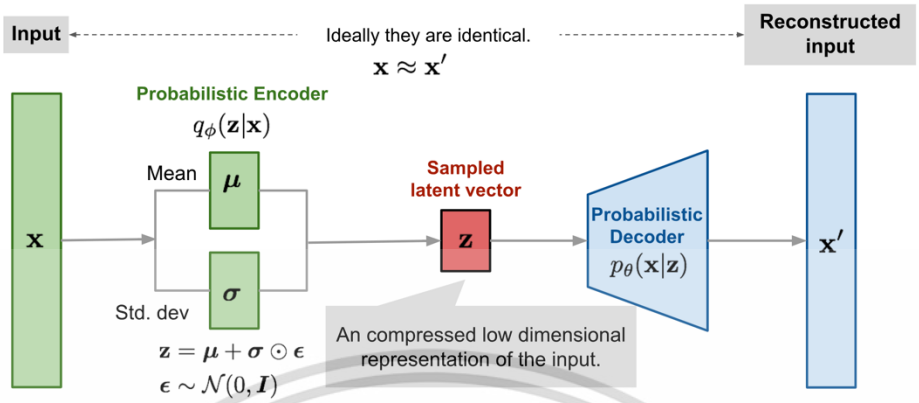


Figure 2.10 VAEs with multivariate Gaussian [70], [73].

2.3.2.2 Implicit Density Model: Generative Adversarial Networks (GANs)

Unlike the explicit density model, the implicit density model does not need to directly measure or fit the data distribution. The implicit density model generates data samples from the distribution without an explicit hypothesis and uses the sample results to modify the model. Generative adversarial networks (GANs) are a class of machine learning that has two simultaneously trained models (neural networks, unsupervised), one is the Generator and another one is Discriminator. The first model uses to generate fake data while the latter is trained to distinguish the fake or the real data. As the name suggests "generative" GANs can produce new data, and "adversarial" represents the duel of those two models which are "neural networks"[74]. These neural networks can be simple feed-forward networks, e.g., multi-layer perceptron (MLP), or more advance like convolutional neural networks (CNNs). GANs were not the first computer code or program that could generate data, however, their synthesizing data and unique and high-quality results, especially in producing super realistic images made GANs separate from the rest. Moreover, GAN needs no labeled data which we know is gold in data science, as it is pricy [66].

The basic idea of GAN is to generate new data by the adversarial process of two networks. GAN is first introduced by [74] and consists of two networks, e.g., the Generative network as G and the Discriminative network as D. These two networks will duel in a minimax two-player game. The training of G wants to

maximize the probability that D is making a mistake in its discrimination. Finally, in arbitrary function space, G can recover the training/input distribution while D is equal to 0.5 everywhere or achieve the Nash equilibrium. In this state, the quality of data generated by G has been like the input so that D cannot distinguish it anymore. The GANs architecture is illustrated in Figure 2.11 and the summary of the Generator and Discriminator subnetworks is provided in Table 2.3.

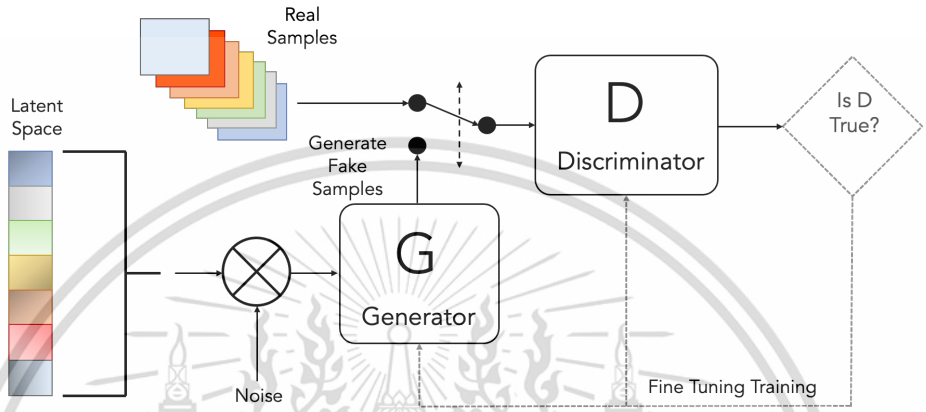


Figure 2.11 GANs illustration (redrawn from [75]).

Table 2.4 G and D subnetworks.

	Generator (G)	Discriminator (D)
Input	A vector of random numbers (noise)	Two inputs: • Real samples from real dataset • Fake samples generated by G
Output	Fake data samples that very similar and as convincing as real data samples.	Give prediction probability that the input samples are true/real.
Goal	Generate fake data that cannot be distinguish to members of training/real dataset.	Distinguish between real samples from dataset and fake samples from G.

GANs Training

Let us assume both networks are in the form of MLP. MLP is a fully connected class of feed-forward neural networks so that the entire GAN can be trained with backpropagation. Compared to other methods, GAN does not need the Markov chain or unrolled approximate of Bayesian inference either in training or generating the sample data [76]. The overall GANs training algorithm can be explained as follows:

GANs Training Algorithm

For each training iteration ***do***

1. Train the Discriminator:
   a. Take a random sample from real dataset as x.
   b. Get random noise vector, z via Generator network then synthesis the fake sample as x'.
   c. Use the Discriminator network to classify between x and x'.
   d. Compute the classification errors.
   e. Backpropagate the total error to Discriminator's trainable parameters for *minimizing* the classification error.
2. Train the Generator:
   a. Get random noise vector, z via Generator network then synthesize the fake sample as x'.
   b. Use the Discriminator network to classify x'.
   c. Compute the classification errors.
   d. Backpropagate the total error to Generator's trainable parameters for *maximizing* the Discriminator's error

End for

Figure 2.12 shows the visualization of GANs training algorithm. GANs reached Nash equilibrium when the Generator achieved a similar level of likeliness of its fake output compared to the real dataset while the Discriminator gave a 50/50

prediction of the input. However, in the real application, reaching Nash equilibrium is quite difficult.

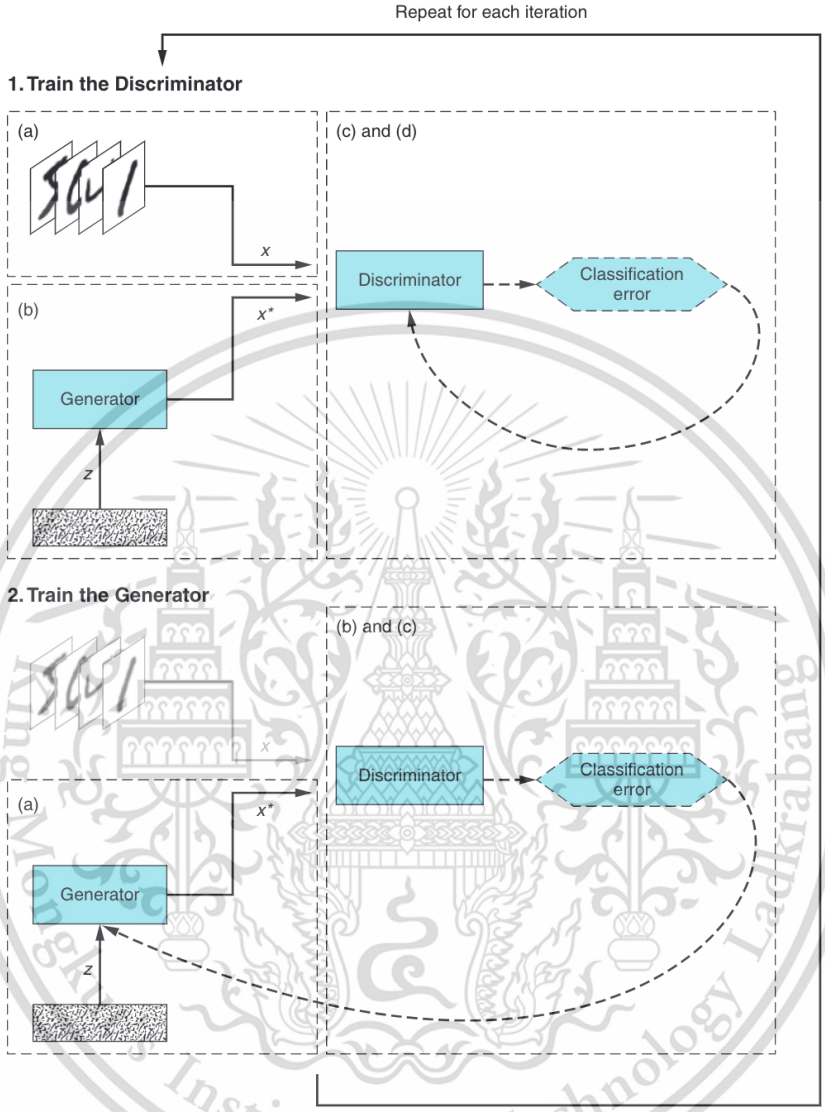


Figure 2.12 Two main parts of GANs training [66].

Adversarial Nets [74]

The easier assumption is that both Discriminator (D) and Generator (G) networks are represented by MLP. The "adversarial nets" process can be described as:

1. To learn the Generator's distribution, p_g over the real/input data, x, we can define a prior distribution of input noise, as $p_z(z)$, z is the noise latent.
2. To represent mapping to the data space, $G(z, \theta_g)$, is a differentiable function by MLP with parameter θ_g.

3. $D(x, \theta_d)$ is the second MLP representing the Discriminator and has only one output scalar. $D(x)$ represents the probability that x comes from the actual data rather than from the p_g.
4. The D is trained to maximize the probability to assign the correct/true label to both training examples and samples from G.
5. Simultaneously from (4), train G to minimize $\log\big(1 - D\big(G(z)\big)\big)$
6. The value function of $V(D, G)$ can be seen as a two-player minimax game as

$$\begin{aligned} &min_G\, max_D\, V\,(D, G) \\ &\quad = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ &\quad + \mathbb{E}_{z\sim p_z(z)}\big[1 - \log D\big(G(z)\big)\big] \end{aligned} \tag{2.11}$$

7. Training criterion allows one to recover the data generating distribution as G and D are given enough capacity, i.e., by a non-parametric limit.
8. $V(D, G)$ in practice is approached by iterative, numerically procedure.

In this manner, we alternate between the k steps of optimizing D and one step optimizing G, resulting in D being maintained near its optimal solution if G is slowly changing. In (1), we can observe that instead of minimizing $\mathbf{1} - \log D\big(G(z)\big)$, we can train to maximize $\log D\big(G(z)\big)$.

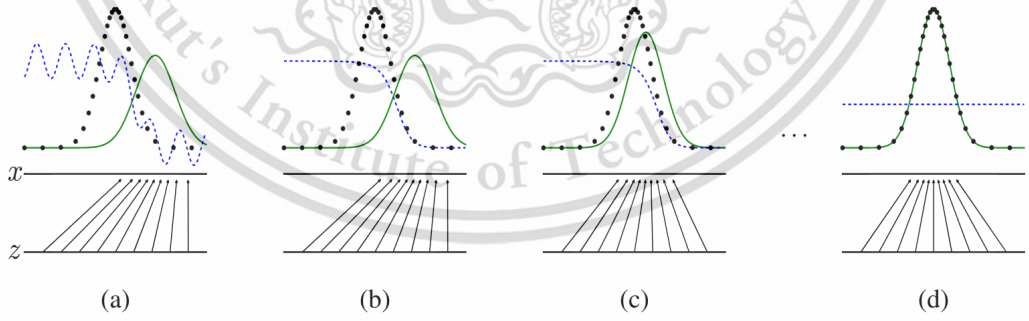


Figure 2.13 Simultaneously training of GANs [74].

- Figure 2.13 depicts the explanation in the point (7). Here, the blue-dashed line is the discriminative distribution, D, the black dotted line is the real sample distribution, p_x, and the green solid line is the generative distribution, $p_g(G)$.

- The lower horizontal is the domain of $\boldsymbol{z}$ sampled uniformly. While the above horizontal, is the domain of $\boldsymbol{x}$.
- The arrow (upward) shows the mapping of $\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{z})$ imposes the non-uniform distribution $\boldsymbol{p_g}$ on the transformed sample.
- $\boldsymbol{G}$ will have high focus in the high density and expanding in the low density of $\boldsymbol{p_g}$.

The four processes from (a) to (d) in Figure 2.11 can be elaborated as:

a) If we consider that our learning is near convergence, p_g is like p_{data} and the Discriminator, D is becoming a good classifier (blue-dashed line).

b) Then, when the D is trained to distinguish from the sample data, converging to $D^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_g(\boldsymbol{x})}$.

c) After updating on G, D's gradient has guided the $G(\boldsymbol{z})$ to flow to regions likely to be classified as data (real).

If these steps continue and if G and D have enough capacity, there is a certain point that both networks cannot improve. Thus, $p_g = p_{data}$ then the $D(\boldsymbol{x}) = \frac{1}{2}$ because it cannot distinguish between those two distributions.

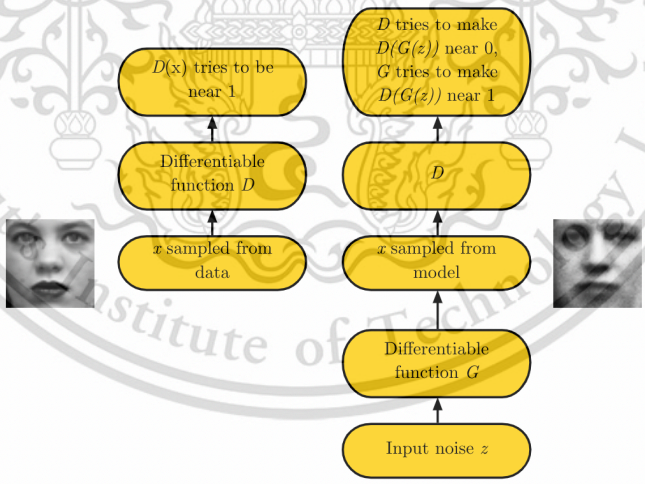


Figure 2.14 The adversarial framework of GANs illustration [65].

2.3.3 Fingerprint Technique via Deep Generative Model: Related Works

In this subchapter, we will present the related works related the DL-based indoor localization. We focused on the use deep generative model, especially for VAEs and GANs implementations on the fingerprint database enhancement method, as DL methods are pervasive. Some proposals for DL method for database enhancement,

including data augmentation [76]–[79]. Table 2.5 summarizes the DL algorithms used for fingerprint-based technique in our research's contextual frameworks.

Table 2.5 Related works on DL algorithms used for fingerprint-based technique.

Author (year)	DL models	Evaluation Frameworks		
		Parameter	Method	Results
Chen et al. (2021) [80]	VAEs, GANs	CSI	Semi-supervised.	GANs have the better performance than VAEs.
Qian et al. (2021) [81]	GP, MDNs, VAEs	RSSI	Semi-supervised VAEs.	Compared to GP, semi-supervised VAEs is less computational complex. While compared to MDNs, VAEs are more stable in learning process.
Li et al. (2019) [82]	GANs	RSRP and RSRQ	Sparsely self-supervised GAN (SS-GAN).	SS-GAN model is superior to baseline models in terms of the average RMSE.
Chen et al. (2022) [83]	VAEs	RSSI	VAEs as data augmenter to introduce data diversity.	Submeter level accuracy. Best on adaptation of diverse unlabeled data dynamic environments.
Alhomayani et al. (2021) [84]	VAEs, conditional VAEs	RSSI	VAEs and c-VAEs are used to produce class-balance dataset in the imbalance dataset issue.	Outperformed SMOTE and ADASYN (Adaptive Synthetic).

Zhang et al. (2021) [85]	Combination (VAEs & GANs)	Indoor SDR data (dB). LTE spectrum (dB), sensor scope (^{0}C)	AAEs handles measurements with large missing regions and complicated loss patterns.	Outperformed the existing methods on missing data.
Njima et al. (2021) [77]	GANs	RSSI	GANs generates fake RSSI and pseudo labeling for its label.	Increase the accuracy of 21.69% for simulated data and 15.36% for the experimental data.
Zou et al. (2020) [86]	GPR with least-squares GANs	RSSI	GPR used to roughly predict the RSS values with its coordinate.	Combination of GPR-GANs can achieve synthetic RSSI values even outside the measurement area.
Nabati et al. (2020) [76]	GANs	RSSI	GANs is used to generate the synthetic data.	By only 10% real + synthetic data from GANs, the accuracy is outperformed baseline methods by using 100% real data.

*Reference Signal Received Power (RSRP), Reference Signal Received. Quality (RSRQ), Gaussian Process (GP), Mixture Density Networks (MDNs), Gaussian Process Regression (GPR), Adversarial Autoencoders (AAEs), root mean square error (RMSE), synthetic minority over-sampling technique (SMOTE).

Chen et. al. [80] proposed the semi-supervised VAEs and GANs for fingerprint-based indoor localization utilizing CSI. They found that GANs have superior performance compared to VAEs. They also discussed the different generative mechanisms and environmental effects by comparing deep generative networks. RSSI is also used for the deep generative-based model, i.e., in [87]. It has a slightly different application, as in this publication, the semi-supervised VAEs are used more to estimate the location than applied as data augmenters. In this paper, the VAEs have benefits compared to GP and MDNs, in terms of complexity and stability in the learning process. In the unique

approach of modified GANs in [82], the authors proposed the sparsely self-supervised generative adversarial nets (SS-GAN). They proposed a novel data-driven model to generate the RF maps of an area from irregularly distributed measurement samples so that the radio maps can be enhanced. The actual measurement campaign on 4G-LTE was conducted, and the results showed the superiority of SS-GAN compared to the baseline model. The datasets are also unique compared to regular RSSI, which in this case, they utilized Reference Signal Received Power (RSRP) and Reference Signal Received. Quality (RSRQ).

Fidora is introduced in [83], which applied VAE for data augmenters. It also provides fast adaptation to new or diverse unlabeled data by applying a domain-adaptive classifier. For the solution in the imbalance dataset, the authors in [84] proposed a combination of VAEs and c-VAEs to produce the class-balance dataset. The application of VAEs and GANs is also beneficially used for missing data in fingerprint-based indoor localization [85]. Using GANs and pseudo labels to generate fake RSSI is proposed by [77]. By applying GANs and generating fake RSSI, the issue of a lack number of training data is tackled, and the accuracy performance is improved. Gaussian Process Regression (GPR) and least-square GANs (LS-GAN) can achieve exciting and accurate RSSI synthesis even if it is outside the area of interest. The basic idea is that instead of using only the noise vector as Generator input, it uses the rough prediction of noise from the RSSI-coordinate values via GPR [86].

Our proposal in deep learning-based indoor localization in this dissertation is more straightforward in implementing a deep generative model to synthesize RSSI data so that the issue of the small and sparse database can be alleviated. A similar method and publication have become one of our primary references [76]. The authors implemented GANs to synthesize the dataset and add the synthetic data to a small fraction of the actual data. By doing so, the proposal only needs 10% of the actual data fraction to surpass the performance of baseline methods employing 100% actual data. From here, it can be concluded that we can reduce the number of actual fingerprint database construction by applying a deep generative model for database enhancement or synthesis. Then, one challenge of fingerprint technique can be tackled.

2.4 Research Gaps

We have investigated the prospective research gaps in each fingerprint-technique database enhancement approach.

2.4.1 Dual Features Fingerprint Database for Indoor Device-free Localization

Indoor device-free localization (IDFL) does not need the person holding the devices for localization. To achieve this purpose, the use of the fingerprint technique is essential. We found that limited references apply dual or more features for IDFL due to its cost and weak implementation points. However, we consider applying dual features, RSSI and illuminance, which can be obtained simultaneously in the measurement campaign, even using low-cost devices [48]. There is also limited use of dual localization parameters straightforwardly and combining with the ML algorithm, e.g., Random Forest and the ML's baseline methods.

2.4.2 Classical Interpolation and Regression

Most of the publications apply interpolation techniques that have some degree of complexity. We need a simple but effective interpolation method based on empirical data. As the author's concern, there is also limited publication discussing the basic interpolation techniques for 2D and 3D scenarios. These two considerations are the research gaps that we explored and proposed.

2.4.3 Deep Generative Models-based Fingerprint Database Enhancement

The small fingerprint database is applicable for reducing the database construction burden issue. However, having only a small/sparse database cannot provide good training data for an ML-based indoor localization system, especially DL implementation. Some published works have discussed several DL algorithms for data augmentation and target location determination. However, most of the databases used for their approach are whether it is the public dataset or their measurement campaign, which is still considered a large dataset. Using our small measurement campaign and applying DL's generative and deterministic are the research gaps that we can fill in this dissertation.

Chapter 3

Database Enhancement via Dual Features Dependent

Fingerprint-based technique usually only use one localization parameter, e.g., received signal strength indicator (RSSI), channel state information (CSI), or others. In this chapter, we discuss how can we add another parameter to make more unique properties on a specific location with its features dependent value.

This chapter is published in Sensors and Materials (S&M) Journal [24]:

> Dwi Joko Suroso, Panarat Cherntanomwong, and Pitikhate Sooraksa, "Indoor Device- free Localization Using Received Signal Strength Indicator and Illuminance Sensor for Random-forest-based Fingerprint Technique," Sensors and Materials, vol. 33, no. 12, pp. 4331–4345, 2021, doi: 10.18494/SAM.2021.3632.

Supplementary data and implementation code in Python can be accessed at https://github.com/dwijokosuroso.

3.1 Abstract

Indoor device-free localization (IDFL) offers more flexibility than the conventional indoor localization (device-based) system, as the targets or objects do not need to carry any device to be located. In IDFL's localization process, the target is passive, and it unlocks some applications, i.e., elderly monitoring, security intruder, and indoor navigation. Despite having more flexibility than device-based, IDFL is still inferior in terms of localization performance. The most used localization technique for IDFL is the fingerprint technique. This technique uses the uniqueness of spatial information to predict the target's location. The spatial information is the fingerprint database containing information of locations and parameters corresponding to those locations. The most specific parameter for the fingerprint database is the received signal strength indicator (RSSI). Without additional hardware installation, RSSI can be obtained directly from many low-cost devices, i.e., Wi-Fi-based devices. The two-phase process in the fingerprint technique; the database is constructed in the offline phase, and the online phase is the matching process to compare the target's current parameter with those in the database. We propose a fingerprint technique-based IDFL by using RSSI and illumination from an illuminance sensor for the additional parameters of the fingerprint database. Both parameters are recorded by considering two scenarios: the empty room and the person standing in the fingerprint grids. The constructed database is the person-filled subtracted to the empty room database. We propose the Random Forest as the pattern matching algorithm. As one of the machine learning (ML) algorithms, we evaluate the Random Forest performance by comparing its performance with other ML algorithms; k-Nearest Neighbor (k-NN) and Neural Networks (NN). The results show that k-NN has better accuracy than the Random Forest for learning and testing by comparing their root mean square error (RMSE). On the other hand, the Random Forest has better accuracy than NN and better precision than both k-NN and NN for learning and testing by observing the standard deviation (STD). The results show that there is the possibility to improve the IDFL performance by adding more parameters in the fingerprint database and using an ML-based pattern matching algorithm.

Keywords: device-free, indoor localization, RSSI, illuminance sensor, random forest, machine learning.

3.2 Introduction and Background

In 2007, Youssef *et al*. introduced the concept of passive indoor localization by using the difference in the received signal strength indicator (RSSI) for communication between Wi-Fi routers with two access points (APs) and monitoring points (MPs). They identified the change in RSSI caused by the existence of a target with detection probabilities of 1 and 0. By applying Bayesian inference, they found that the detection accuracy was 86% when they implemented a fingerprint-based technique [41]. Passive indoor localization, explicitly known as indoor device-free localization (IDFL), is expected to have more flexibility and applicability [88]–[90]. Unlike device-based localization, in IDFL the target or person does not need to carry a device. The IFDL procedure will likely use the difference in the environment conditions received by a communication device installed in the surrounding area [91]. Radio-frequency-based technologies, i.e., Wi-Fi [92], Zigbee standard [93], radio frequency identification (RFID)[94]–[96], Bluetooth low energy (BLE)[97], [98] have been most commonly used for IDFL. These technologies provide the signal parameters applied to localization techniques [2], [16]. Some signal parameters are easy to obtain without additional hardware installation, whereas others are complex, sophisticated, and costly. RSSI is widely used in indoor localization [99]–[101] as it is straightforward to use, compatible with the above technologies, and has a low cost [102]. However, RSSI is prone to signal fluctuation and multipath interference in relatively dense indoor environments [103]. Although it has been claimed that channel state information (CSI) based on the channel characterization and channel model has superior performance to RSSI, its complexity and the cost of measurement are high [25], [104], [105].

We can select a suitable indoor localization technique to tackle the drawbacks of RSSI. Two main techniques are used in device-based indoor localization. The first is based on range- or distance and the signal parameters are converted to distances, followed by the localization process. The second is range-free, and spatial information consisting of location information is collected and the corresponding signal parameter in the same location is stored. In IDFL, the range-free technique is most commonly applied, i.e., the fingerprint technique, as it is impossible to convert the signal to the distance between the target and reference points without attaching a device to the target or person. On the basis of this argument, we propose an RSSI-based IDFL system. RSSI from a Wi-Fi device is preferable because of its low cost and its high availability in almost all smart devices.

The essential point of the fingerprint technique is the quality of the database. The fingerprint technique employs a two-phase process. The first process is called the offline phase, where the database fingerprints are recorded and stored at a specific grid location in the area of interest, which is the area in which the indoor localization system is set up. The second phase is the online phase, in which a new target parameter is acquired from the target or object and compared with those in the database by applying a pattern-matching algorithm. This two-phase process is often used in device-based systems. The IDFL fingerprint process is slightly different in that it builds a "passive" offline database [90], [106]. Seifeldin *et al*. proposed fingerprint-based IDFL in which a probabilistic model was applied. A continuous space estimator based on the RSSI vector was used as the location estimator. The median distance error was 1.82 m. Even though the estimation accuracy was low, their method was superior to deterministic and random-estimator-based methods [107].

Other IDFL approaches are to apply radio tomographic imaging [93], [108], [109] and a lighting infrastructure, i.e., LED sensing [110]–[112]. One requirement for RTI is the employment of many reference sensors/nodes, which is not cost-effective. On the other hand, the lighting infrastructure usually can only predict the existence of a target or object and not its location. Other illumination-based approaches use modified lamps in a system that is not easily recreated or have an unrealistic lamp installment on the floor. These are some disadvantages of applying illumination-based IDFL [112].

How a fingerprint-based IDFL system stores the fingerprint database has been explained in some previous papers. A key feature of this system is that the parameters stored, i.e., RSSI, are in the form of RSSI values when the area is empty and when a person is standing at a certain position in the designed grid. The difference in the RSSI values is stored as the fingerprint database. In the online phase, the target, without a device attached, wanders inside the area of interest. This causes changes in the RSSI values received by the reference, from which the position can be predicted on the basis of the similarity to the database. Several proposals for the pattern-matching algorithm, including the use of machine learning (ML) and deep learning, have been discussed [17], [113]. Some ML algorithms can work well with sparse data, such as the decision tree and random forest algorithms [114], [115]. The random forest has been demonstrated to have high accuracy in some proposed indoor localization applications with relatively small datasets.

We propose a new approach for passive fingerprint databases where we combine RSSI and illuminance data by installing an illuminance sensor along with a communication device. Using two or more parameters for the fingerprint technique in IDFL is still uncommon. Furthermore, techniques based on sensor fusion are probably more common. However, the disadvantage of sensor fusion is the computational complexity of finding the weights of particular sensing parameters [17], [116], [117]. Therefore, we use another parameter to provide spatial information for the fingerprint technique by utilizing the attached illuminance sensor and processing the obtained information using the device that exhibits and receives the RSSI. We expect that by adding more parameters to the fingerprint database, the unique spatial information for different locations will markedly differ. We utilize the random forest algorithm as our pattern-matching algorithm and compare its localization performance with two other ML algorithms: k-nearest neighbor (k-NN) and neural networks (NN).

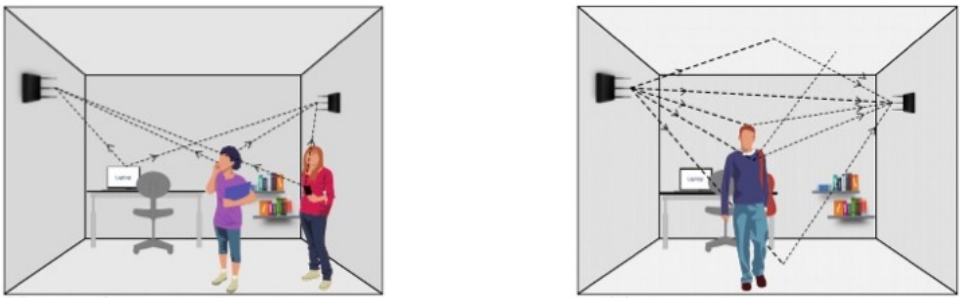
We have so far introduced the context of our research. Section 2 comprises a discussion of IDFL, the use of the random forest as the pattern-matching algorithm, the measurement setup, and the employed performance metric. The results and discussion are presented in Sect. 3. Finally, in Sect. 4, we conclude our work and outline our planned future work.

3.3 Materials and Methods

Compared with device-based indoor localization systems, IDFL requires more reference nodes to obtain a spatial signature in the area of interest. In this section, the primary difference between device-based and device-free systems, the fingerprint-based techniques for IDFL, the random forest, our measurement setup, and the proposed performance metric are explained.

3.3.1 Device-based vs. Device-free System

IDFL is used to detect, track, or identify a target, object, or person without the need for the target to carry a localization device, in contrast to the device-based localization method, where the target must be equipped with a specific tool such as a smartphone or an electronic tag. Figure 3.1 shows the difference between device-based and device-free localization [118].



(a) (b)

Figure 3.1 Indoor localization illustration (a) device-based (b) device-free [10].

In contrast to device-based localization, which utilizes measurement parameters captured by a device placed on the target to identify the target position, device-free localization utilizes changes in measured parameters in the surrounding environment resulting from the target's interaction with surrounding objects. For example, RF-based IDFL technology utilizes various phenomena caused by radio signal propagation, such as absorption, scattering, diffraction, reflection, refraction, or a combination of these phenomena.

3.3.2 Fingerprint-based Technique for IDFL

A passive fingerprint database is different from a device-based fingerprint database. A passive fingerprint database is collected based on the data of RSSI values difference between the RSSI values measured from an empty room and those measured with a person/object inside the area of interest. Figure 3.2 illustrates passive fingerprint database collection. The environmental change due to the presence of the person causes a difference in the receiver power, e.g., RSSI. We measured the discrepancies in the RSSI values between the cases of an empty room and of a person/object inside the room and used them to create a fingerprint database.

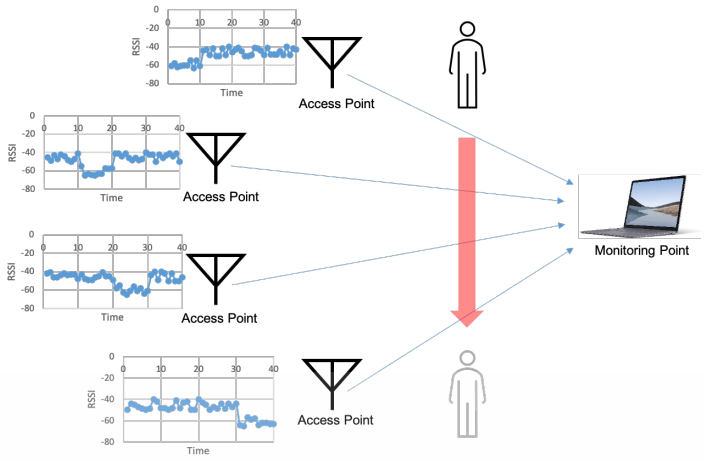


Figure 3.2 Passive fingerprint database collection.

First, we must design the fingerprint location grid in the measurement location. Then, following the two-phase fingerprint technique, we record RSSI in the empty room and in the room with a person standing in the designed grid for the offline phase. We create an IDFL database of data of the difference between the RSSI values obtained under the two conditions. The new RSSI values from the present target are then subjected to a pattern-matching algorithm in the online phase. Similar to the process of constructing the fingerprint database, these new RSSI values are also subtracted from the RSSI values for the empty room.

After the comparison, the fingerprint location with RSSI values having the highest similarity to the new RSSI values indicates the target's predicted location. In this study, we propose the addition of illuminance values as a parameter for the fingerprint technique to offer greater uniqueness than that obtained using only RSSI. Unlike the difference between the RSSI values of the empty and person-filled room, the illuminance values are used straightforwardly, and illuminance values in the database are compared with new values obtained when a target is localized. Figure 3.3 illustrates the process of the IDFL fingerprint technique.

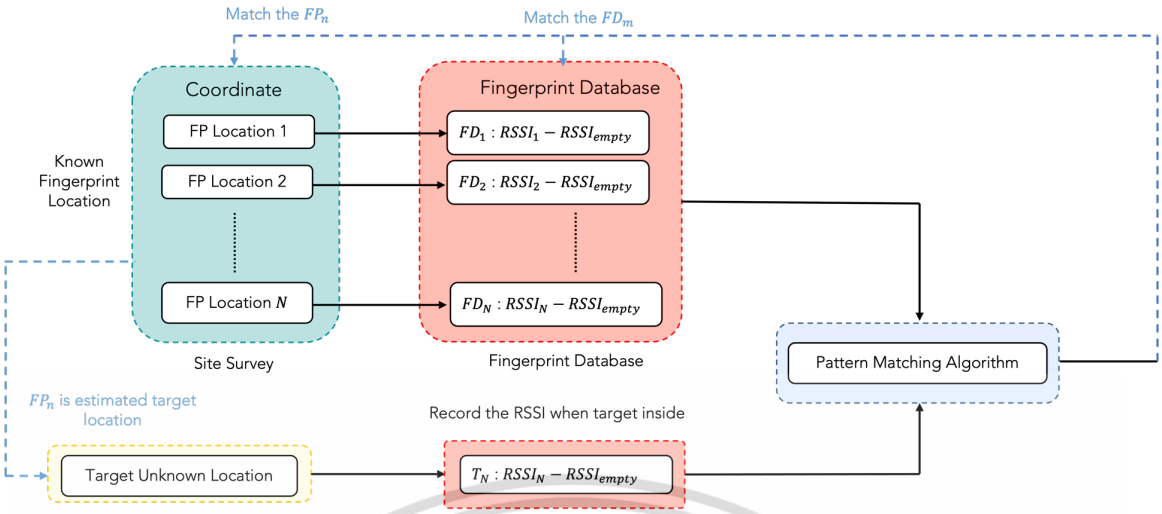


Figure 3.3 RSSI-based IDFL illustration process.

3.3.3 The Random Forest

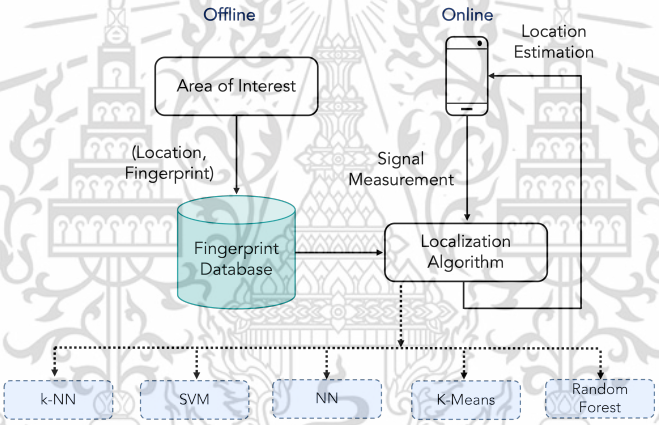


Figure 3.4 ML-based pattern matching algorithm[17].

ML-based pattern-matching algorithms are now commonly used thanks to advances in computing technology and natural programming language. Figure 3.4 shows the most widely used ML-based pattern-matching algorithms, i.e., k-NN, NN, support vector machine (SVM), and k-means clustering algorithms. The random forest is an extension of the decision tree algorithm. The structure of the decision tree algorithm resembles a tree, where the root node consists of all the training data [119]. Then, the root node is divided into two or more child nodes on the basis of a particular condition. A child node is an internal node that either can be split further or cannot be split and becomes a terminal node. A specific value is used to label the terminal node. Child nodes are repeatedly split until the stopping criterion is satisfied.

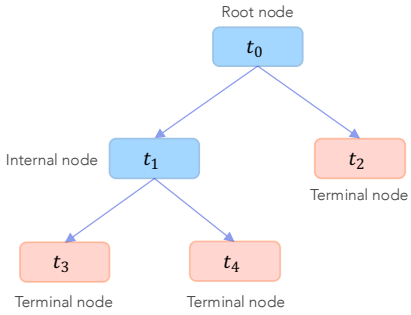


Figure 3.5 Decision tree algorithm illustration.

Random forest is a combination of decision trees, where each decision tree is trained using different but an equal amount of data. Random forest uses the bootstrap aggregating (bagging) technique and random variable selection to build each decision tree. The bootstrap technique is used to make different training datasets by removing some data from the original training dataset and replacing it with the remaining data randomly [120], [121]. Each decision tree is combined by taking the most popular class for the classifier and averaging every prediction by each tree for the regressor (aggregation). Aggregation acts as a classifier when it calculates the more significant votes as a result and as a regressor by calculating the mean values of each tree's prediction results. Random forest offers high accuracy and is suitable for fingerprint-based IDFL with a relatively small dataset. This algorithm is also free from overfitting because it takes the mean of all predictions in the process. Figure 6 shows the general flow of random forest for prediction.

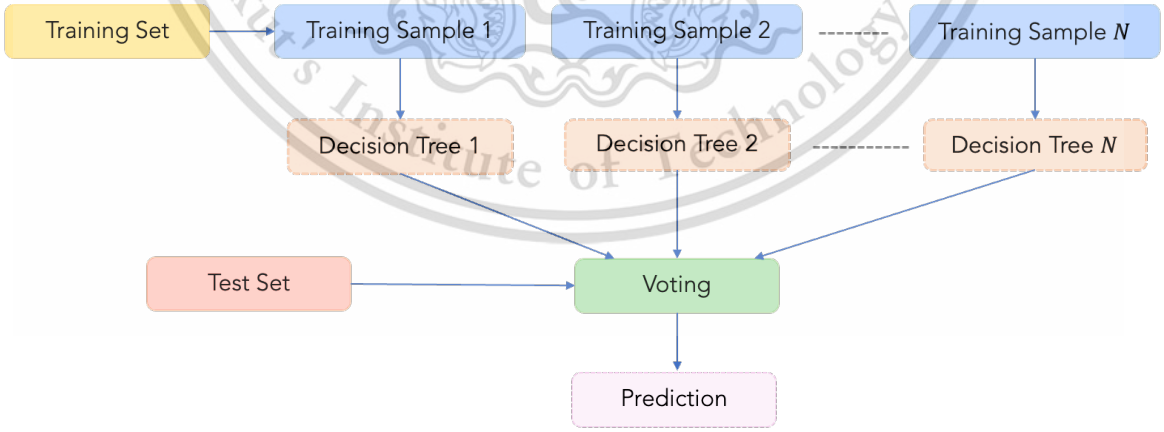
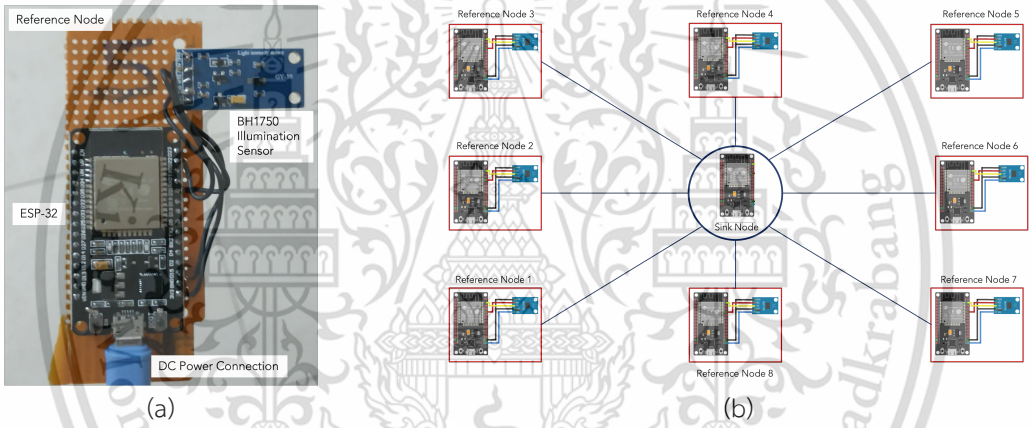


Figure 3.6 Random Forest prediction process.

3.3.4 Measurement Setup

We used Wi-Fi-based ESP32 devices as the core of our system for both the reference nodes and the sink node. The reference nodes acted as action points (APs) that broadcast the RSSI values continuously; once each reference node becomes the receiver, it will receive all RSSI values from other APs and measure the cumulative illumination from its position. Seven RSSI values from other reference nodes or APs and one illuminance value are sent to the sink node. The process is repeated for all eight references (APs). A reference node consisted of an ESP32 device, a BH1750FVI illuminance sensor, and a DC power connection. To ensure portability, we used a power bank for each reference node. Figure 3.7 shows the actual reference node and the arrangement of the devices in the communication topology.



(a) (b)

Figure 3.7 Measurement system (a) Actual setup (b) Reference and sink arrangement.

Table 3.1 Measurement campaign details.

Device (tools)	Specification (properties)	Note (details)
Reference node	ESP32 Devkit SoC	Support : Wi-Fi IEEE 802.11 b/g/n dan BLE, Memory: 520 KiB SRAM
Illuminance sensor	Digital Ambient Light Sensor BH1750FVI	Measurement Range =1-65535lx, Sensor Type : Photodiode with A/D, Power Supply: 2.4V-3.6V
Power	Power bank	> 7000 mAh
Software	Ardunio IDE	1.8.5 version (64-bit)

Algorithm	Jupyter Notebook	Python 3.7 with library Scikit learn, Numpy, Matplotlib, Pandas, and Keras [122], [123]

Figure 3.8 depicts the layout of the measurement setup for the proposed method, and Fig. 3.9 shows the actual measurement environment. Our setup in a classroom had four 16 W LED tubes as the lamp/light source. The reference node with the red arrow indicates that the ESP32 antenna is facing in that direction. There was light leakage from the windows in the actual measurement area, but we covered the windows with fabric to minimize the light noise. There was a Wi-Fi router inside the classroom, from which we expected interference. However, we assumed that the recorded values of both RSSI and illuminance in the database included these effects.

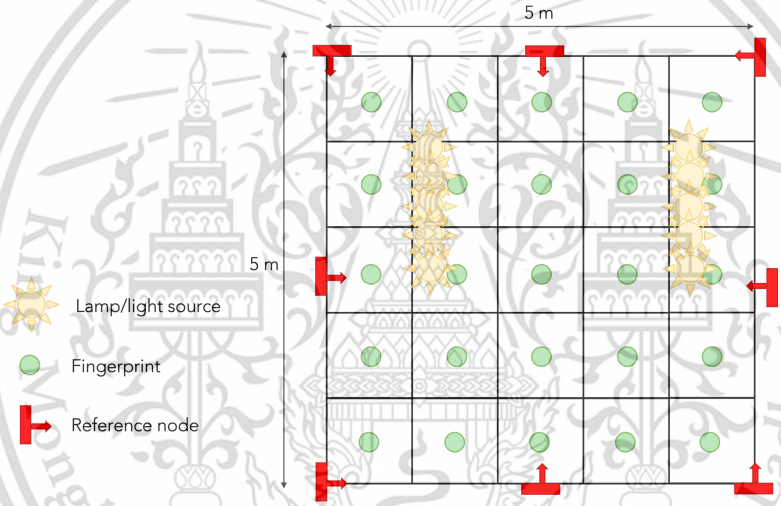


Figure 3.8 Measurement campaign layout.

(a) (b)

Figure 3.9 Measurement setup (a) Empty room (b) A person in dedicated fingerprints.

We constructed database as follows. In the first procedure, we recorded the RSSI and illuminance values in an empty room by sequentially switching a reference node to act as a receiver and collecting the RSSI values from the other seven reference nodes. The node that acted as a receiver also collected the cumulative illuminance at

its position and sent it to the sink node/server. The second procedure was the same, but a person stood at a fingerprint location (one of the green circles in Fig. 3.8) then moved to the next location after 2 min of RSSI and illuminance data collection. Table 2 shows the data structure received by the sink node, and the acquired database is expressed as [Eq. (3.1)].

Table 3.2 Measurement data structure.

Node ID	RSSI 1	RSSI 2	RSSI 3	RSSI 4	RSSI 5	RSSI 6	RSSI 7	RSSI 8	Lux	RS*

*reading stamp (RS).

$$RSSI_{fingerprint,i} = \left|RSSI_{person,i} - RSSI_{empty}\right|, \left|lux_{person,i} - lux_{empty}\right| \quad (3.1)$$

$RSSI_{fingerprint,i}$ with $i = 1,2,\ldots,25$ is the database corresponding to each fingerprint, where RSSI data obtained with a person standing at the fingerprint location is subtracted from the measured RSSI data of the empty room. We ultimately obtained a database of data for 25 fingerprint locations, for each fingerprint location, the RSSI values and the illuminance values measured for 2 min by the illuminance sensor. A similar process was carried out to obtain the target data, $RSSI_{target,i} = \left|RSSI_{target,i} - RSSI_{empty}\right|, \left|lux_{target,i} - lux_{empty}\right|$, where lux_i is illumination values in lux for the i^{th} fingerprint and the corresponding target location.

3.3.5 Performance Metric

We consider accuracy and precision to validate our IDFL system performance. Accuracy is represented by the root mean square error (RMSE) between the predicted and actual positions, [Eq. (3.2)], while precision can be evaluated from the standard deviation (STD) of the distribution of predicted data points [Eq. (3.3)].

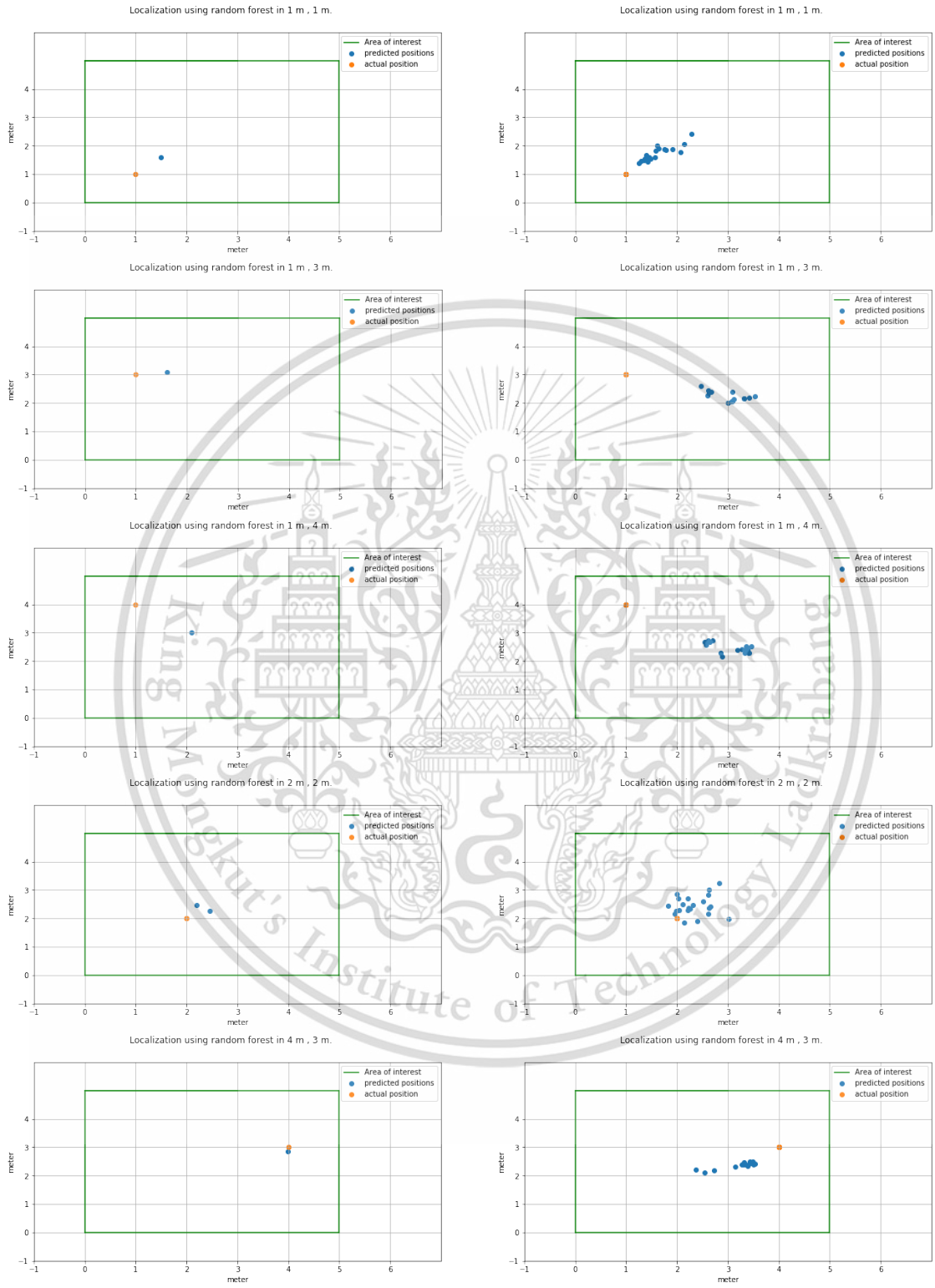
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((x_{predict,i} - x_{actual,i})^2 + (y_{predict,i} - y_{actual,i})^2)} \quad (3.2)$$

The precision of predicted location points can be observed by the standard deviation, defined as Eq. (3.3).

$$STD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}((x_{predict,i} - x_{actual,i})^2 + (y_{predict,i} - y_{actual,i})^2) - RMSE)} \quad (3.3)$$

3.4 Results and Discussion

3.4.1 Random Forest Localization Performance



(a) learning (b) testing

Figure 3.10 Localization result using Random Forest (a) Learning (b) Testing.

We present the localization performance first as the results of learning using the fingerprint database. As we have test data (not real-time online phase), 99% of the data was used for learning and 1% for testing. Second, we performed a test with real data to evaluate the performance of the proposed method by observing its accuracy and precision. We showed the results only for some positions with interesting learning and testing results. The learning showed some errors, but they were less than 1 m. However, the testing results showed some large errors. In the target area (1 m, 1 m), the RMSE of the learning result was 0.77 m and that of the testing result was 0.96 m.

Figure 3.10 shows that the testing results have high variance, as shown by the STD, compared with the learning results. This may be because the test data have more variance than the learning data. However, visual inspection revealed that the predicted locations in the testing scenario are distributed near the true/actual location. Testing of another location yielded similar results, i.e., for target location (2 m, 2 m), the testing results were distributed close to the actual location. From these results, we observed that the testing results tended to be distributed in the middle of the area of interest. One of the reasons may be that the RSSI is easily affected by environmental effects and signal interference.

3.4.2 Performance Comparison

Figure 3.11 shows the accuracy represented by RMSE and precision represented by STD of the learning results. For learning, the data in the database was divided into the training dataset and the learning/pretesting dataset. RMSE of random forest is worse than that of k-NN, as it was 0.8 m while it was a lower value of 0.25 m for k-NN. NN has the highest RMSE of 0.93m. A similar trend was seen for STD of random forest compared with k-NN and NN. However, note that both RMSE and STD results for random forest were still under 1 m (measurement grid is 1 m).

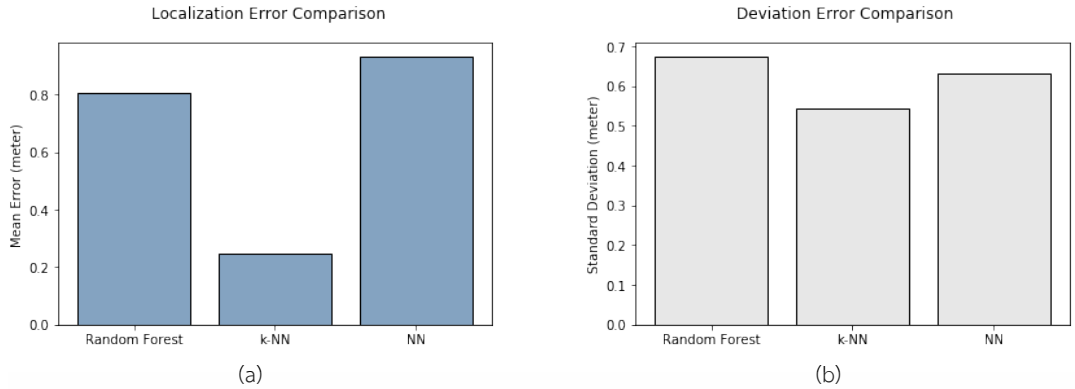


Figure 3.11 Training performance comparison (a) RMSE (b) STD.

We compared the performance of random forest with those of k-NN and NN in terms of RMSE and STD. RMSE indicates the accuracy of the localization result, and STD shows the precision of the results of location prediction for a specific location. Figure 3.12 shows RMSE and STD of random forest, k-NN, and NN for localization using actual test data. As shown in Fig. 3.12(a), random forest is slightly inferior to k-NN, as its mean localization error is 1.65 m compared with 1.46 m of k-NN, but that of NN is 2 m. Figure 3.12(b) shows that the precision of random forest is superior to those of the two other algorithms, indicating that the predicted values have relatively low variance, the STD from the actual position being 0.88 m.

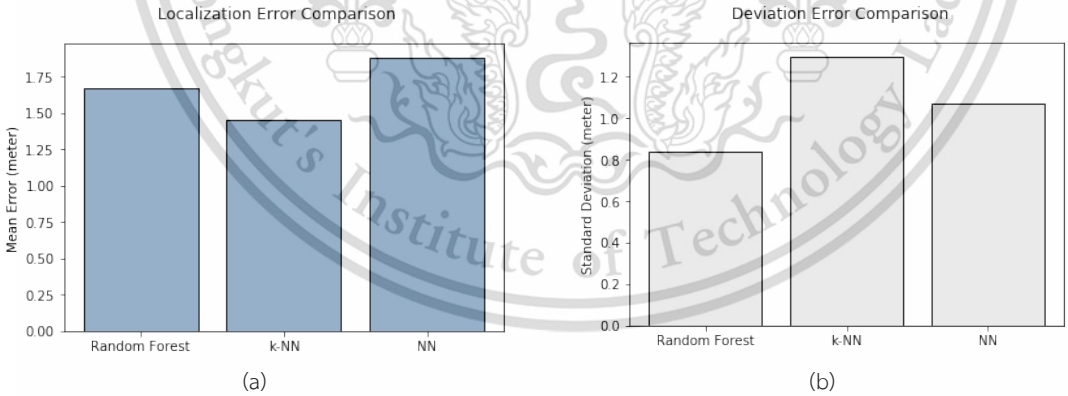


Figure 3.12 Testing performance comparison (a) RMSE (b) STD.

3.5 Conclusion

We presented an IDFL system utilizing RSSI and illuminance values for fingerprint localization and applied the random forest algorithm for pattern matching. From the localization training and testing results, random forest was found to have better precision, represented as STD, than other ML algorithms, i.e., k-NN and NN. However, k-NN was slightly better than random forest in terms of localization accuracy represented by RMSE. The localization results indicated that the overall performance is still relatively low. Unlike the device-based localization system, where the parameter values for fingerprint databases show larger differences, the effects of RSSI fluctuations on the IDFL system are similar between an empty and an occupied space. However, there is room for improvement by, for example, applying another signal parameter such as CSI, improving the data collection method, adding a number of datasets, and implementing deep learning in our future work. We expect that implementing deep learning and the use of CSI, which is more robust and reliable than RSSI, will improve the performance of IDFL [124].

Chapter 4

Database Enhancement via Classical Interpolation

Classical interpolation is straightforward and yet there are still open opportunities to employ it as the fingerprint database enhancement. Especially the application in 2D and 3D environment scenario.

This chapter is based on our publication in Emerging Science Journal (ESJ) Journal [18]:

> Dwi Joko Suroso, Farid Y.M. Adiyatma, Panarat Cherntanomwong, and Pitikhate Sooraksa, "Fingerprint Database Enhancement by Applying Interpolation and Regression Techniques for IoT-based Indoor Localization," Emerging Science Journal, vol. 4, pp. 167–189, Jan. 2022, doi: 10.28991/esj-2021-sp1-012.

Supplementary data and implementation code in Python can be accessed at https://github.com/dwijokosuroso.

4.1 Abstract

Most applied indoor localization is based on distance and fingerprint techniques. The distance-based technique converts specific parameters to a distance, while the fingerprint technique stores parameters as the fingerprint database. The widely used internet of things (IoT) technologies, e.g., Wi-Fi and ZigBee, provide the localization parameters, i.e., received signal strength indicator (RSSI). The fingerprint technique advantages over the distance-based method as it straightforwardly uses the parameter and has better accuracy. However, the burden in database reconstruction in terms of complexity and cost is the disadvantage of this technique. Some solutions, i.e., interpolation, image-based method, machine learning (ML)-based, have been proposed to enhance the fingerprint methods. The limitations are complex and evaluated only in a single environment or simulation. This paper proposes applying classical interpolation and regression to create the synthetic fingerprint database using only a relatively sparse RSSI dataset. We use bilinear and polynomial interpolation and polynomial regression techniques to create the synthetic database and apply our methods to the 2D and 3D environments. We obtain an accuracy improvement of 0.2m for 2D and 0.13m for 3D by applying the synthetic database. Adding the synthetic database can tackle the sparsity issues, and the offline fingerprint database construction will be less burden.

Keywords: indoor localization, IoT, ZigBee, fingerprint technique, fingerprint database, interpolation, regression, polynomial.

4.2 Introduction and Background

Internet of Things (IoT) technology advancement has been flourishing in the last decades [125], [126]. Its implementation cannot be excluded from our everyday life. One of the features that are frequently used is positioning. As the well-established global positioning system (GPS) is most used for positioning, especially in the outdoor environment, it fails to give the proper accuracy positioning in the indoor environment [16], [127]. Therefore, some IoT-based technologies, i.e., Wi-Fi [3], [128], ZigBee [129], Bluetooth Low Energy (BLE) [98], Ultra-wideband (UWB) [16], Radio Frequency Identification (RFID) [5], can be applied for Indoor Positioning Systems (IPS) instead of GPS's utilization

indoor [12]. More commonly stated as indoor localization, IPS can be achieved by applying the technologies mentioned above and specific methods or techniques.

Wireless sensor networks (WSNs), also based on IoT technology, are known as one of the most used for indoor localization implementation [16], [130], [131]. This paper considers the low-cost and straightforward implementation of WSNs-based indoor localization using the ZigBee standard [21], [131], [132]. Compared to other technologies such as Wi-Fi, the ZigBee can have a more flexible setup and deployment of the WSNs system. On the other hand, indoor localization techniques include algorithms to identify the target's location based on several signal properties based on the technologies offered.

ZigBee standard provides some signal properties that can be used for localization algorithm, i.e., Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) [113], [133]–[135]. The advantages of RSSI include simple to obtain and straightforward implementation without additional hardware installation. However, the drawback of RSSI properties is that the signal somehow fluctuates and varies due to time. For LQI, the additional hardware to record the data transmission may require, and its implementation is not typical [135].

The indoor localization techniques are generally divided into two, as the range or distance becomes the center of concern. The distance-based and the distance-free techniques are widely applied for indoor localization techniques [7]. Distance-based techniques solely rely on distance measurement to estimate the location. This distance can be from the conversion parameter of signal power, i.e., RSSI mentioned before, or angle or time information from the signal transmission. One advantage of this range-based technique is that if the algorithm has received an excellent distance-signal properties conversion, the localization can be done straight away without pre-processing or databasing. These advantages also depend on the kind of signal properties used for distance conversion. However, the range-based technique will yield a high error if unreliable signal properties are received and poorly converted as the distance parameter [136].

The distance-free indoor localization technique, on the other hand, has advantages in reducing the effect of this signal fluctuation caused by the multipath effect in the indoor environment by collecting the spatial information of this signal to record as the database. The common distance-free technique in indoor localization is fingerprinting [137]. The fingerprinting technique requires two phases for the localization, the first phase is called the offline phase, and the second phase is the online phase [138]. In the offline

phase, the necessary information related to the spatial information of the area of interest is recorded as the offline fingerprint database. The fingerprint database consists of the signal properties information related to a specific location on the designated grids inside the area of interest: the denser these grids, the more accurate and precise later for the online phase or the localization process. After the data or localization parameters are measured and stored in the database acquisition, the online phase, in which the target or object sends the signal parameter to the system and receive the same parameters as recorded in the database, the pattern matching algorithm will work by comparing the target's signal parameters to those in database. This algorithm then concludes that the target or object belongs to a particular position with similar spatial information recorded in the fingerprint database.

However, the fingerprint technique has drawbacks in the offline database construction phase [17], [139]. It takes much effort, high cost, and sometimes has complexity or scalability issues when the applied indoor environment is enormous. There are several attempts to reduce the drawbacks, i.e., constructing the artificial grids by applying classical to machine learning-based techniques and reducing the database complexity by employing some compression algorithm [140]. The effort in signal point-of-view enhances the signal parameters using the dedicated filter to fight the signal fluctuations and implements several clustering techniques to remove the data outliers and improve localization accuracy [141], [142]. For all of this method, as the author is concerned, there are still open challenges in implementing accurate and straightforward fingerprint database enhancement, i.e., simple interpolation and regression techniques [56], [58].

Some proposals that applied the interpolation techniques to tackle the database sparsity have been published in [59], [60]. The most used interpolation technique for fingerprint-based indoor localization is the Kriging technique. The authors in [33], showed that the Kriging for RSSI, especially in inaccessible areas, can be covered and enhance the overall fingerprint database. Another approach of the database enhancement by path-loss model-based interpolation is available in [58], [61], the crowdsourcing method [62], Spatio-temporal similarity [143], and clustering-based and interpolation on [144]. However, most of these approaches have limitations on algorithm complexities. Furthermore, some issues related to the advancement parameter, e.g., channel state

information (CSI) and its complexity, appear and become the drawbacks of applying this parameter.

Thus, as far as our concerns, the simple yet straightforward implementation of interpolation and regression technique is not yet considered—especially when using WSNs-based as the system's core. By utilizing RSSI following the log-loss distance, it views the linear assumption relationships between signal strength and the distance. By considering this, the interpolation and regression can be established as the power-distance relationship in the applied environment. In addition, we consider conducting an actual measurement campaign in our approach. The approach is the algorithm development in simulation or theory and actual implementation both for two-dimensional (2D) and three-dimensional (3D) by using the ZigBee standard as the core of the WSNs system. Our original achievements are enhancing the density database by using a relatively sparse actual measurement database by applying basic interpolation and regression. Our approach is relatively simple compared to previously mentioned publications. Our significant difficulties to overcome is the RSSI fluctuation in some parts of fingerprint position because of the nature of the environment, i.e., near the edge, enormous metal material, and unbalanced obstruction in the environment, making the synthetic database challenging to assure.

The fingerprint technique for both environments is applied, and the offline database fingerprint is obtained by the area of interest 5×5 m2 for the 2D environment. We utilize the bookshelf as the 3D environment, assuming several floor applications in a multi-story building in the same environment as the 2D settlement. We design the database grid of 1 × 1 m for the 2D and 22×12.5×35 cm for the 3D environment. The cm-scale is to observe how our interpolation and regression algorithm can work well and acceptable in such a short distance. The pattern matching used is the classical minimum Euclidean distance (MED) to know-how is the difference between the predicted and the actual RSSI values. The tiniest error yield from the target-database comparison is assigned as the target's location. From the context we have explained and by proposing the method for actual implementation, we would like to highlight the contribution of our proposed method as follows:

1. Implementing the interpolation and regression technique for database enhancement in fingerprint-based indoor localization considering 2D and 3D environments.

2. We compare the interpolation and regression techniques to observe the best use of the technique for the case of 2D and 3D environments.

We present the structure of this paper as follows; in the first part, we discuss the introduction and background stated the context and importance of our proposal. In the second part, we present the indoor localization technologies and techniques and the fingerprint technique's comprehensive explanation. For the third part, we detail the material and method, including the ZigBee-based WSNs system, interpolation, regression technique we propose to implement, pattern matching, performance metrics, and the measurement system and setup. Results and discussion will then be presented in the fourth fifth part. Finally, we will conclude our findings and discuss our proposal limitations and plan of our near-future works.

4.3 Indoor Localization and Fingerprint Technique

4.3.1 Indoor Localization

Indoor localization can act as an "indoor GPS," and some technologies have been introduced to develop indoor localization research. The technologies include radio frequency (RF)-based and other, i.e., mechanical, optical/light wave, acoustic wave, and vision-based, also attract the research further. However, RF-based research is most common and widely implemented. Table 1 shows the most-used RF-based indoor localization technologies and their features [7].

Table 4.1 RF-based indoor localization technologies.

Technology	Accuracy (meter)	Range (meter)	Power (watt)
GPS	1 - 20	global	500
RFID	1	1 - 50	0.02 - 0.3
Wi-Fi	1 - 5	<100	0.5 - 1
UWB	<0.3	<300	0.03

BLE	1	<10	0.001
ZigBee	1 - 5	<30	0.02 – 0.04
GPS	1 - 20	global	500

In this paper, we consider applying the ZigBee technology because of several reasons:

- By comparing the accuracy of other technologies in Table 1, the ZigBee system can achieve an accuracy of 1 – 5 m. UWB technology can achieve better accuracy than ZigBee. However, implementation of UWB need additional hardware, and to achieve <0.3 m accuracy, several advances and complicated algorithm are needed, e.g., Time-of-Arrival (ToA), Ranging Time (RT).
- Observing the range properties of ZigBee technology, the <30 m can cover almost all indoor environments, the scalability issues can be solved.
- Power consumption is also low compared to other technologies; only RFID and BLE give the same or less than ZigBee. However, the range of BLE is relatively short, while the RFID technology also requires the line-of-sight (LoS) communication between tags and the RFID reader.

ZigBee standard is the IEEE 802.15.4 standard working in industrial, scientific, and medical (ISM) bands similar to Wi-Fi in 2.4 GHz. In the measurement system and setup, we will explain how to reduce the effect of the signal interference with Wi-Fi.

4.3.2 Distance Measurement Technique

Figure 4.1 shows the several distance measurement or signal properties used in indoor localization.

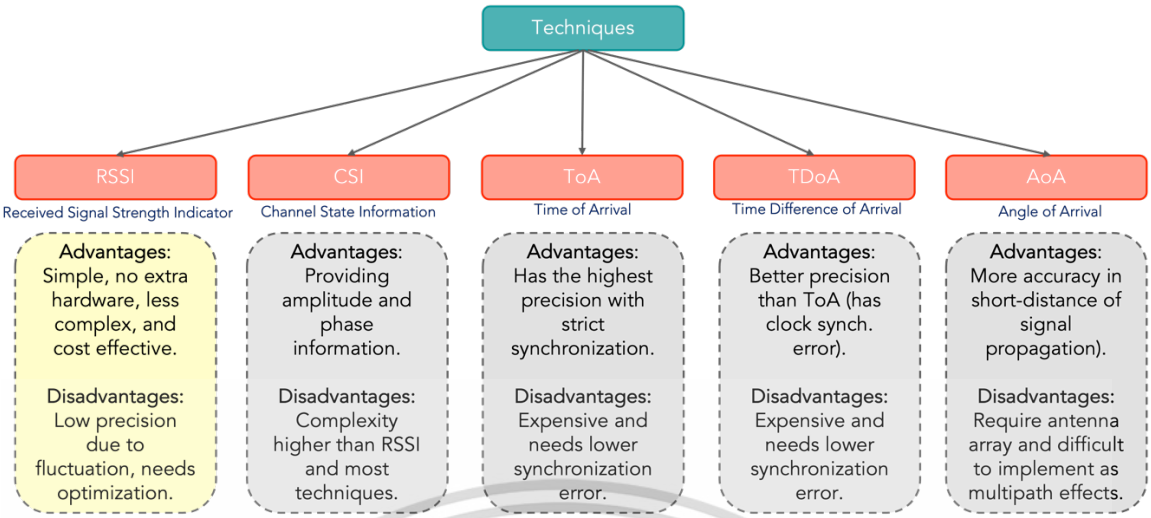


Figure 4.1 Distance measurement technique.

There are some advantages and disadvantages of each distance measurement technique. However, we can note from the illustration that the RSSI parameter has the most prospective implementation, especially in our approach. First, it has advantages in the straightforward implementation, the second there are disadvantages point that becomes a focal point, especially for the database enhancement in our proposal. RSSI is defined as the received power, and it follows the log-loss distance model [145]:

$$RSSI(dBm) = A - 10 \cdot n \cdot log_{10}\left[\frac{d_0}{d}\right]. \quad (4.1)$$

RSSI in dBm related to the distance, d, is equal to the A, power in dBm of reference distance, d_0 subtracted by the path loss exponent, n, multiply the log distance of d_0 divided by the d; generally, d_0 is 1 m. The values of n can be measured by empirical in the particular indoor environment used. In our approach, we collected the RSSI values from the ZigBee standard without considering the RSSI-distance conversion.

4.3.3 Indoor Localization Methods

Figure 4.2 shows several basic methods applied in indoor localization systems.

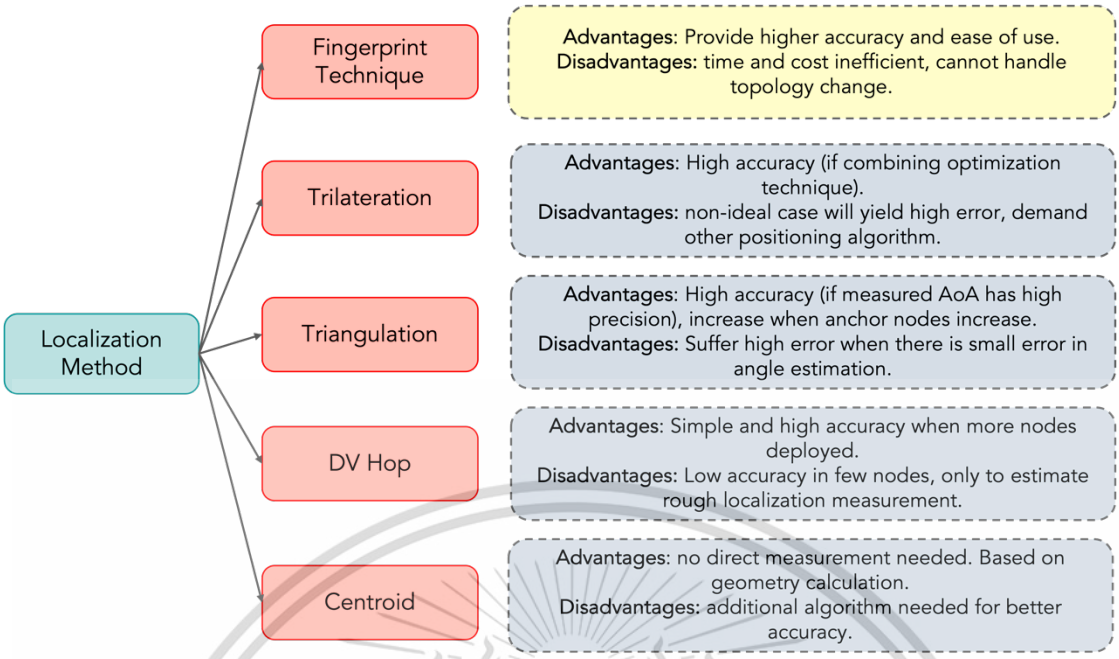


Figure 4.2 Localization methods.

The fingerprint technique gives advantages when there is no need for distance-parameter conversion. Furthermore, the fingerprint technique gives better accuracy than several range-based techniques, i.e., triangulation or trilateration, min-max, and iRingLa compared based on RSSI as localization parameters [146]. The other techniques can also yield high accuracy but need additional support in the algorithm, precise clocking, accurate angle estimation, and the high number of nodes or beacons used for positioning.

For previously mentioned advantages of fingerprint technique, however, there is a drawback of fingerprint technique, especially in the offline database construction process, with the burden of the cost and time inefficient, moreover, if there is an application in the large-scale indoor environment, the system will be more complex and needs more human resources. The trade-off of this technique is that if we do not have enough density of fingerprint database, the localization error will be high. On the contrary, when we need to have a very dense fingerprint database, multiple drawbacks will appear related to offline databasing. In this paper, we propose the database enhancement in tackling the drawbacks of the fingerprint technique, especially in the database sparsity, by applying the interpolation and regression technique.

4.3.4 Fingerprint Technique

The fingerprint technique is similar to fingerprint pattern recognition in image processing. However, the terms fingerprint here refers to radio fingerprints' spatial information. In the previous section, we discussed the general definition of the fingerprint technique. We have also shown the disadvantages in the fingerprint technique implementation, especially in the database construction process. This section will discuss other related fingerprint techniques, their disadvantages, and solutions. The illustration of the fingerprint technique process can be depicted in Figure 4.3.

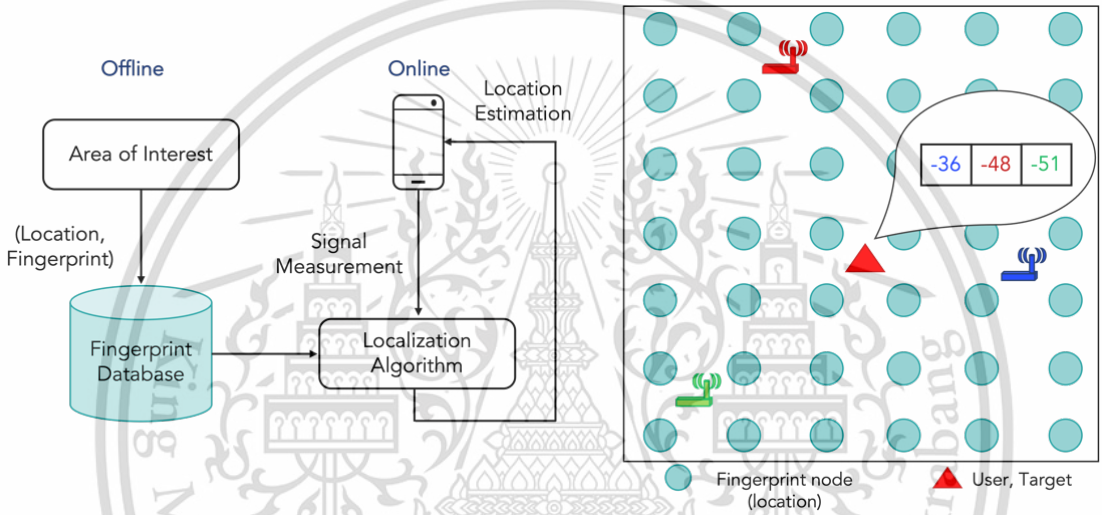


Figure 4.3 Fingerprint technique illustration.

The fingerprint technique works by constructing the database with spatial information needed to estimate target or object position. This spatial information is the location of the fingerprint point with its corresponding parameter. Figure 4.3 depicts the corresponding parameter is RSSI values from three reference nodes/beacons. The fingerprint technique applies the two phases; the first phase is the offline phase for storing the spatial information as the database, and the second is the signal measurement by an object to be localized by a localization algorithm or pattern matching algorithm. For our proposal, the detail of our fingerprint technique workflow is depicted in Figure 4.4. The known fingerprint location depicted in Figure 4.3 shown in the RSSI-based fingerprint technique in Fig. 4.4 as the $FP\ Location\ 1, FP\ Location\ 2, \dots, FP\ Location\ N$ will collect the RSSI from the reference nodes i to M, with $i = 1,2, \dots, M$ as the $FD_i, \dots, FD_M$. After all, the

fingerprint points and their corresponding RSSI values are stored in the database. The online phase is when the target calculates the RSSI values from the same i to M nodes and comparing to those in database by applying a pattern matching algorithm. In our proposal, we propose applying the minimum Euclidean distance (MED) algorithm to find the similarity between target parameter, T and the $FD_i, \ldots, FD_M$ in the database. Once the i^{th} fingerprint database, which has similar RSSI values, is successfully concluded, the correspondence location of this i^{th} database, the i^{th} FP Location is estimated as the target or object location.

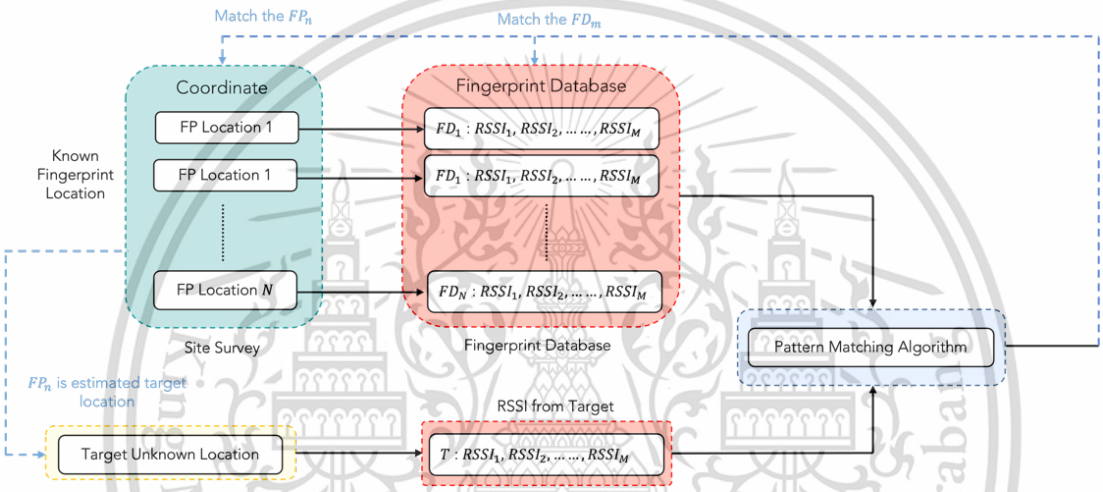


Figure 4.4 RSSI-based fingerprint technique illustration.

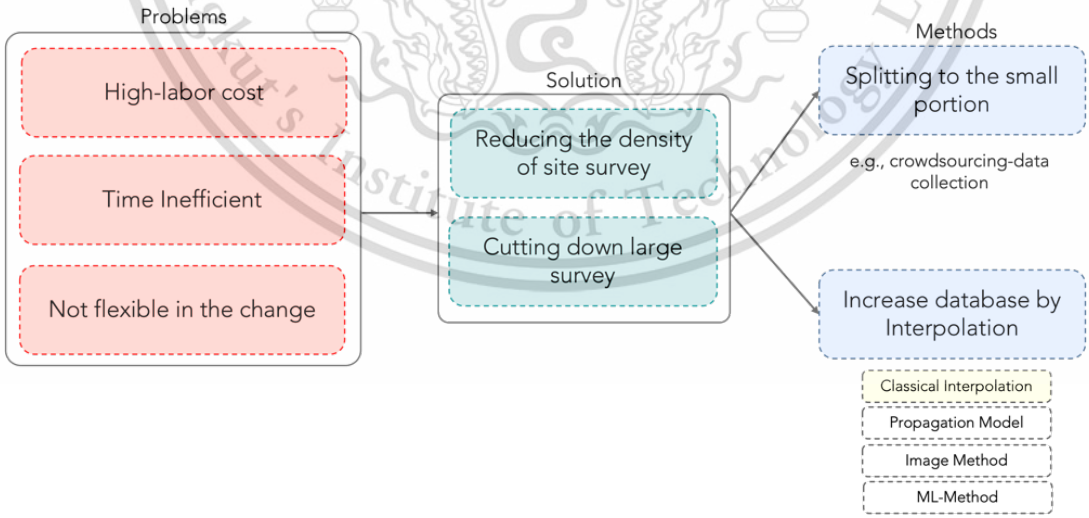


Figure 4.5 Challenges and prospective solutions of the fingerprint technique.

The fingerprint technique performance metrics, i.e., accuracy and precision, are heavily dependent on the quality of the database. One of the factors is the density of

the database. However, there will be more burden processes in the offline phase to get a denser database. Some challenges and their prospective solutions can be seen in Figure 4.5. Cutting down extensive surveys or the database's density improvement can be one of the solutions [147]. Some proposals on the propagation model-based database enhancement have a drawback in the high cost of the channel measurement device and measurement campaigns [148], [149]. The other approaches are by using the image method in which the grayscale image of the image processing method is applied to convert the RSSI values in the area of interest [150]. The imaging method is promising but highly complex in the super-resolution image conversion from the RSSI to grayscale image conversion.

The ML-based method for database augmentation has abundantly proposed, and because of the threshold in the number of data required in some ML techniques, the complexity is also high [17], [104], [151]. As far as the author is concerned, there are few, or there is no attempt yet in using the classical interpolation technique for this database synthesis. Therefore, this paper proposes utilizing the classical interpolation and regression technique with low complexity and achieving acceptable performance results.

4.4 Materials and Methods

4.4.1 Wireless Sensor Networks (WSNs) using ZigBee Standard

The ZigBee device, XBee-24ZB, is utilized for both the target and reference nodes in our WSNs setup. We apply the topology star for the localization system. Here, the sink node is the target located inside the area of interest, while the reference nodes are in the corners of the area of interest. Figure 4.6 shows the illustration of our WSNs based on ZigBee [7], [152].

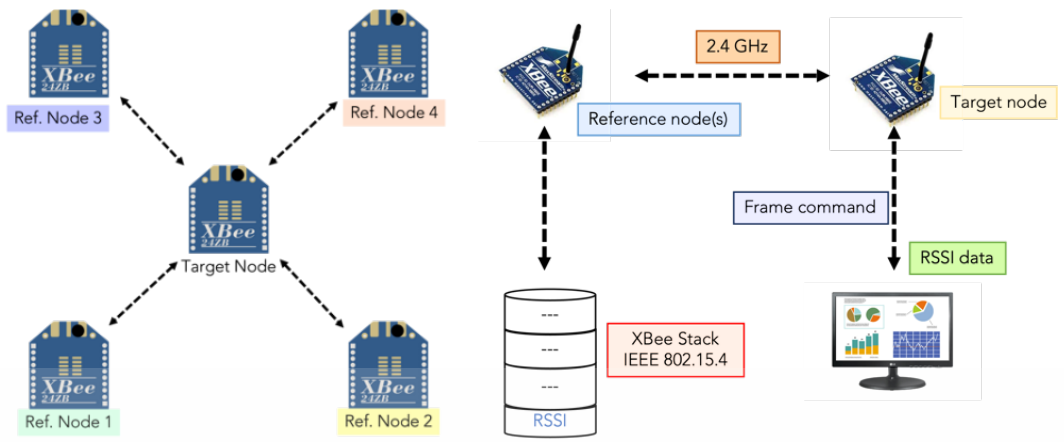


Figure 4.6 a) Topology star illustration, b) the RSSI acquiring process.

The RSSI acquiring process starts with request from the sink node/target to the reference nodes. The reference nodes then get the RSSI packet request by sending back the RSSI from the XBee stack to the sink node. The sink node then translates the packet and store the RSSI values to the storing device/personal computer.

4.4.2 Interpolation and Regression Technique

The interpolation technique predicts a point or a value between two or more known points/values. The basic interpolation techniques are divided into linear and polynomial interpolation. The linear interpolation technique predicts the values between two points, while interpolation can predict specific points using several known data points.

4.4.2.1 Bilinear Interpolation

- The reference of bilinear interpolation is available in [153]. Take example of a simple linear interpolation for two data points, y_1 and y_2, as in Figure 4.7.

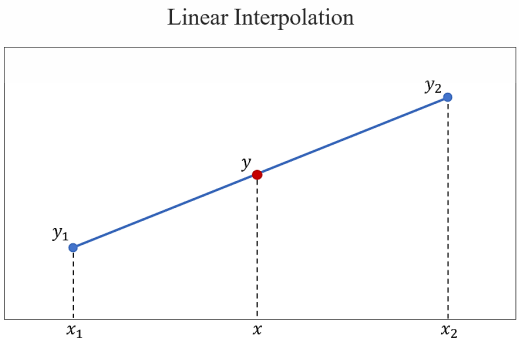


Figure 4.7 Linear interpolation illustration.

The y between y_1 and y_2 can be predicted as in Equation 4.2.

$$y = \frac{x - x_1}{x_2 - x_1}(y_2 - y_1) + y_1 \tag{4.2}$$

- Bilinear interpolation extends a linear interpolation on two cartesian coordinate axes, the x-axis and the y-axis. Suppose there are 4 points (x_1, y_1), (x_2, y_1), (x_2, y_2), dan (x_1, y_2), having functions of $f(x_1, y_1)$, $f(x_2, y_1)$, $f(x_2, y_2)$, and $f(x_1, y_2)$. We can use these values to predict the $f(x, y)$ value at (x, y) as illustrated in Figure 4.8.

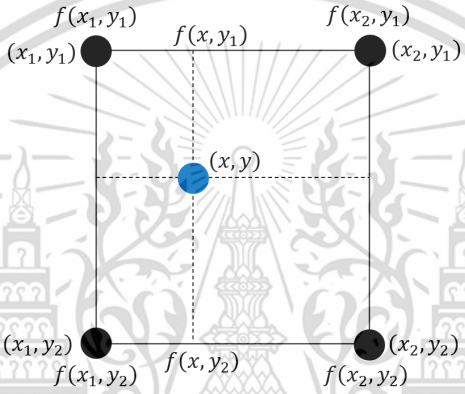


Figure 4.8 Bilinear interpolation illustration.

The value at (x, y) can be predicted by first, we first interpolate the x-axis as

$$f(x, y_1) = \frac{x - x_1}{x_2 - x_1}\big(f(x_2, y_1) - f(x_1, y_1)\big) + f(x_1, y_1). \tag{4.3}$$

$$f(x, y_2) = \frac{x - x_1}{x_2 - x_1}\big(f(x_2, y_2) - f(x_1, y_2)\big) + f(x_1, y_2). \tag{4.4}$$

Thus, by using Equations 4.3 and 4.4, we can the proceed to y-axis interpolation by Equation 4.5.

$$(x, y) = \frac{y - y_1}{y_2 - y_1}\big(f(x, y_2) - f(x, y_1)\big) + f(x, y_1). \tag{4.5}$$

4.4.2.2 Polynomial Interpolation

- The concept of polynomial interpolation is referred to [154].
- If we have more than two data points, polynomial interpolation can predict the data points within the data range. Polynomial interpolation can also handle non-linear pattern data. Neville interpolation is one of the polynomial interpolation methods.

- Predicted value within the data range can be obtained using general equation shown in Equation 4.6.

$$P_k[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{(x-x_{i+k})P_{k-1}[x_i,x_{i+1},\dots,x_{i+k-1}]+(x_i-x)P_{k-1}[x_{i+1},x_{i+2},\dots,x_{i+k}]}{x_i-x_{i+k}}. \quad (4.6)$$

We can solve Equation 4.6 by using Neville Method Settlement detailed in Table 4.2 and k is the degree of the polynomial equation.

Table 4.2 Neville Method Settlement.

	$k=0$	$k=1$	$k=2$	$k=3$
x_0	$P_0[x_0]=y_0$	$P_1[x_0,\ x_1]$	$P_2[x_0,\ x_1,\ x_2]$	$P_3[x_0,\ x_1,\ x_2,\ x_3]$
x_1	$P_0[x_1]=y_1$	$P_1[x_1,\ x_2]$	$P_2[x_1,\ x_2,\ x_3]$	
x_2	$P_0[x_2]=y_2$	$P_1[x_2,\ x_3]$		
x_3	$P_0[x_3]=y_3$			

There are three solution steps for solving the Neville interpolation:

1. **First step:** Performing the interpolation of degree 1 by using the Equation 4.7.

$$P_1[x_0,\ x_1] = \frac{(x-x_1)P_0[x_0]+(x_0-x)P_0[x_1]}{x_0-x_1}. \quad (4.7)$$

2. **Second step:** Performing the interpolation of degree 2 by using the Equation 4.8.

$$P_2[x_0,\ x_1,\ x_2] = \frac{(x-x_2)P_1[x_0,x_1]+(x_0-x)P_1[x_1,x_2]}{x_0-x_2}. \quad (4.8)$$

3. **Third step:** Performing the interpolation of degree 3 by using the Equation 4.9.

$$P_3[x_0,\ x_1,\ x_2,\ x_3] = \frac{(x-x_3)P_2[x_0,x_1,x_2]+(x_0-x)P_2[x_1,x_2,x_3]}{x_0-x_3}. \quad (4.9)$$

4.4.2.3 Polynomial Regression

Polynomial regression is a multiple regression with one independent variable [155]. In one variable polynomial regression equation shown as Equation 4.10, x is expressed as an independent variable.

$$y = a_0 + a_1 x_i + a_2 {x_i}^2 + \cdots + a_n {x_i}^k + e_i,\ i = 1, 2, \ldots, k. \tag{4.10}$$

4.4.3 Pattern Matching and Performance Metrics

We applied a simple Minimum Euclidean Distance (MED) to match the RSSI of the target with those in the database. To validate our localization system's accuracy, we used average distance error (ADE) from all errors of target positions in each scenario, both for 2D and 3D environments [5]. Nevertheless, first, we evaluate the RSSI error value between the actual and predicted RSSI form interpolation and regression techniques by Equation 4.11. This error will be presented in the results and discussion for only 2D, as the RSSI maps from 2D will be easier to interpret. The absolute symbol shows the RSSI discrepancy on the values (because RSSI in dBm is in a negative form). The RSSI error is considered in some points of the 2D fingerprint database to ensure that the proposed method is visible to apply by observing these errors.

$$RSSI\ error\ (dBm) = |RSSI\ value(dBm)_{actual} - RSSI\ value(dBm)_{predicted}|. \tag{4.11}$$

MED as pattern matching utilizes the Euclidean distance to measure the distance from two points; in our case, the Euclidean distance is the distance of RSSI values in the database and RSSI values of the target. The tiniest error or the minimum error of RSSI by their Euclidean distance for specific fingerprint location is assigned as the target's location.

$$Euclidean\ distance\ (m) = \sqrt{\left(RSSI_{database} - RSSI_{target}\right)^2}. \tag{4.12}$$

After we obtain the predicted location based on the similarity of the target and database by the MED, we evaluate the accuracy by the error in the meter of the predicted and the actual target location. Thus, the system's performance metric in our proposal is how the error of the target location prediction, $x_{predicted}$ and $y_{predicted}$, is compared to the actual target location, x_{actual} and y_{actual}. Suppose there are

L number of target locations, we utilized ADE as the performance metric, the mean value of all Euclidean distances for all target locations. If the position of the target node as $i = 1,2, \ldots., L$, we can express the ADE as in Equation 4.13 [156].

$$ADE = \frac{1}{L} \sum_{i=1}^{L} \sqrt{\left(x_{actual,i} - x_{predicted,i}\right)^2 + \left(y_{actual,i} - y_{predicted,i}\right)^2}. \quad (4.13)$$

4.4.4 Methodology Flow Diagram

We design the measurement by applying wireless sensor networks (WSNs)-based data collection. For the flow diagram, we divide into two flows; first flow is an interpolation confirmation step where some of the database points are replaced by the interpolated RSSI values, and the second flow is the interpolation implementation step where the synthetic database from interpolation and regression are combined and evaluated by the localization performance. We start with the confirmation step, where the preparation is to design the measurement for two-dimensional (2D) and three-dimensional (3D) environments, including how we propose the grids of fingerprint and target locations in the area of interest.

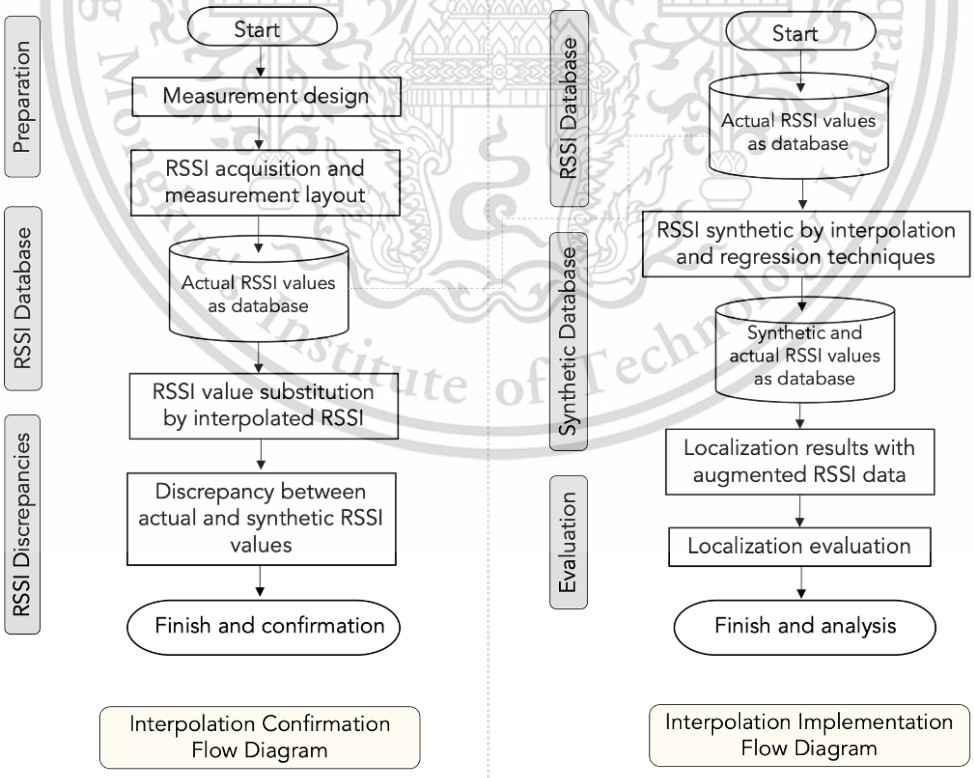


Figure 4.9 Flowchart of Research Methodology.

The RSSI data acquisition step uses the personal computer connected to the sink node (acts as the target node) and follows the star topology. We obtained the actual RSSI and stored it as the fingerprint database. The next step is to do interpolation in the same grid points as the substituted database to ensure the feasibility of using the interpolated data. After confirmation, the second flow uses the actual RSSI values for the database. We applied classical interpolation and regression methods to augment the denser data. We combine the actual and the synthetic database, then evaluate the localization results using the substituted RSSI in some points between and outside (in the case of the 3D environment) of the fingerprint database. The primary step is to make the database denser by applying interpolation and regression as our proposal. We evaluate and compare the localization results for interpolation and regression techniques for the synthetic database as we have the denser database. We discuss the results and conclude what we find.

4.4.5 Measurement Campaign

We consider the two 2D and 3D environments for indoor localization system deployment. The first environment is the lobby environment in our department, and we consider only a 2D scenario. For the 3D environment, we simulate the actual multi-story building by using a 4-level bookshelf for elevation parameters. The ZigBee devices are configured as reference nodes and target nodes; both acted as transceivers and were configured as a star topology.

4.4.5.1 2D Measurement Campaign

We conducted a measurement campaign for the 2D environment in the lobby of our building for a 5 × 5 m2 area of interest. The layout and the actual setup of the measurement campaign for the 2D case are shown in Figure 4.10. We consider using four reference nodes in a 2D environment; this selection assumes that each reference node will be located in each corner of the rectangular shape measurement area. Thus, the interpolation techniques will likely have a better linear relationship between each reference node and its corresponding distance. We have applied more reference nodes for fingerprint-based in the same measurement area and can be found in [21], [152], [157]. Some results suggested that more reference nodes yield better accuracy

(scalability). However, our previous publications do not consider interpolation techniques.

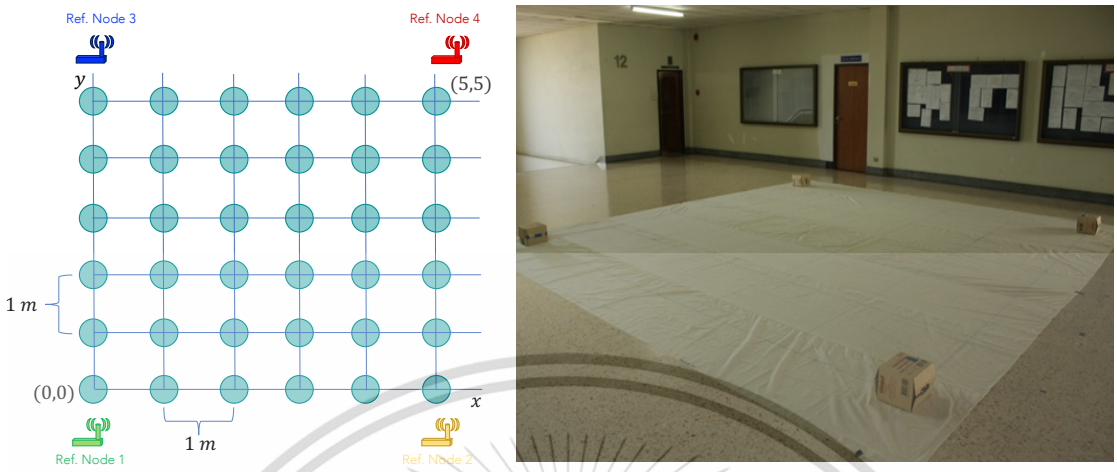


Figure 4.10 2D measurement campaign, a) layout illustration, and b) actual setup.

The four reference nodes are deployed in each rectangular shape's corner, as depicted in Figure 4.10. The measurement grid is 1×1 m2; we have 36 fingerprint locations measurement for the database construction in 5×5 m2 area of interest. The measurement area size has been validated since the RSSI values from the XBee-24ZB device are best in under 8-9m. The 5×5 m2 is still relatively line-of-sight for the lobby itself to support our approach to ensure that the relationship between signal strength and distance is log-loss or linearly decreased. The interpolation techniques need the linear relationship, which is the distance vs. signal strength. ZigBee standard works in the center frequency of 2.4 GHz, the same band with Wi-Fi. To minimize the interference between signals, we turned off all Wi-Fi access points in the measurement area while doing the measurement. The target node is statically placed inside the area of interest. Some positions are the same positions with the fingerprint locations to ensure the quality of the database, some others between two or in the center of four fingerprint locations. The detail of target locations and target type position is shown in Figure 4.11.

The diagonally, vertically, and horizontally placement are considered because of the placement of four reference nodes shown in Figure 4.11a. The effects of the distance and orientation of the target with the reference nodes can be explored by these target positions setup.

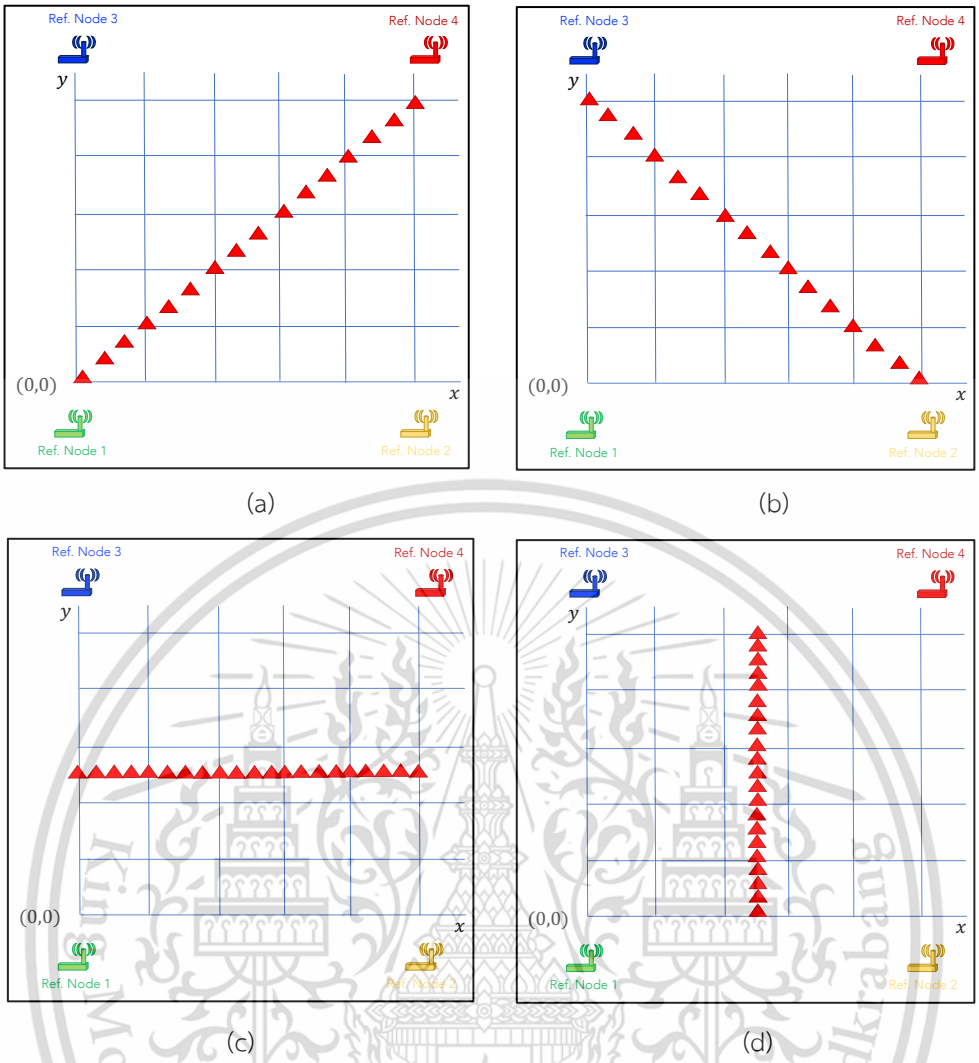


Figure 4.11 Target types, a) type 1, b) type 2, c) type 3, d) type 4.

4.4.5.2 3D Measurement Campaign

To simulate the multi-story building with different levels/floors, we used a four-level wooden bookshelf. We also considered three scenarios in this 3D environment: clean environment, human body effects, and interference objects of books. The body effect is considered by a standing person in 1m distance from the bookshelf. The bookshelf's dimensions are 92 × 25 × 152 cm^3 and the detailed dimension and illustration are depicted in Figure 4.12.

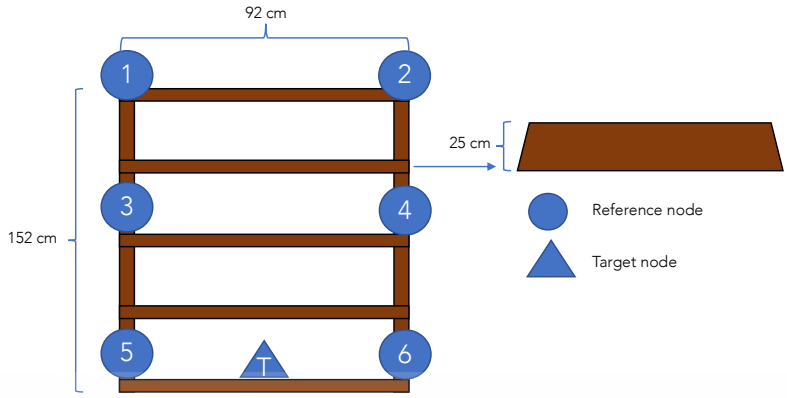


Figure 4.12 Simulated 3D environment illustration in a bookshelf.

The fingerprint node (FP node) and target node can be illustrated in Figure 4.13. Unlike in the 2D environment, in the 3D environment, we employed six reference nodes. This selection ensures that the elevation properties are well represented as the XBee antenna directional pattern is more horizontally shaped. Moreover, we have only six reference nodes in the dataset's measurement setup and have not yet published the approach. The target, T is placed in the different levels of the bookshelf statically.

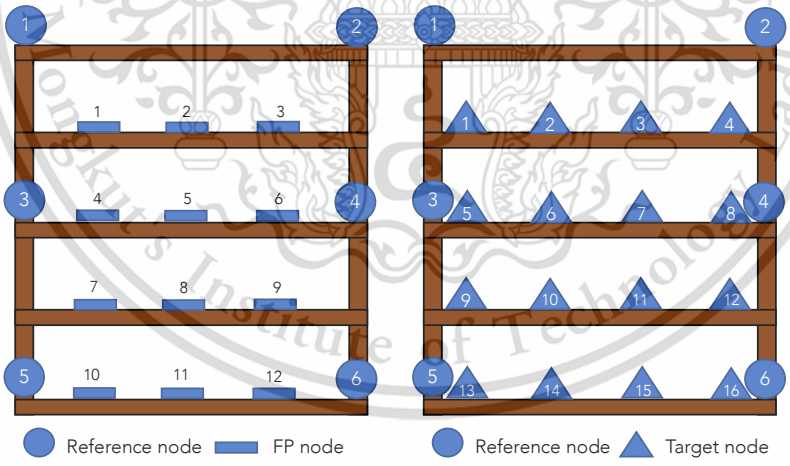


Figure 4.13 Simulated 3D environment, a) fingerprint and b) target positions.

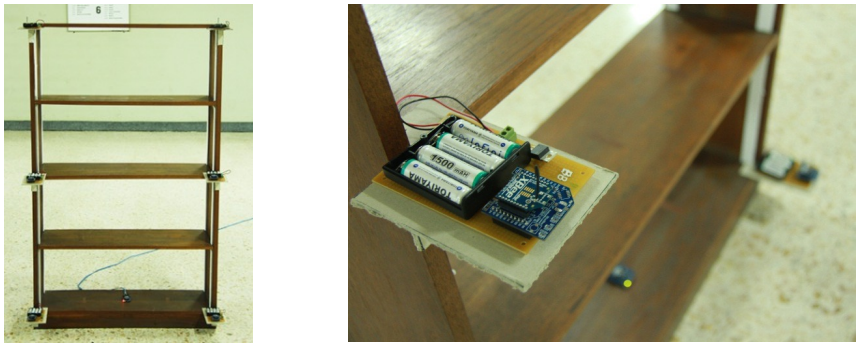


Figure 4.14 The actual setup of 3D environment in a bookshelf.

The 12 FP node locations are used to collect the RSSI values for the fingerprint database. Similar to the 2D environment, here, we also use XbeeXBee-24ZB as the core of the system. The fingerprint grid size is 22 × 12.5×35 cm^3, for length, width, and elevation. The total target node position is 16, located between the FP nodes or near the reference nodes. Figure 4.14 shows the actual placement of XBee-24ZB for the 3D environment setup. The coordinate of fingerprint and target position in the 3D environment is presented in Table 4.3 (FP = fingerprint).

Table 4.3 3D fingerprint and target coordinates.

FP Node	Coordinate (x, y, z)	Target Node	Coordinate (x, y, z)
1.	22 cm, 12.5 cm, 105 cm	1.	11 cm, 12.5 cm, 105 cm
2.	44 cm, 12.5 cm, 105 cm	2.	33 cm, 12.5 cm, 105 cm
3.	66 cm, 12.5 cm, 105 cm	3.	55 cm, 12.5 cm, 105 cm
4.	22 cm, 12.5 cm, 70 cm	4.	77 cm, 12.5 cm, 150 cm
5.	44 cm, 12.5 cm, 70 cm	5.	11 cm, 12.5 cm, 70 cm
6.	66 cm, 12.5 cm, 70 cm	6.	33 cm, 12.5 cm, 70 cm
7.	22 cm, 12.5 cm, 35 cm	7.	55 cm, 12.5 cm, 70 cm

8.	44 cm, 12.5 cm, 35 cm	8.	77 cm, 12.5 cm, 70 cm
9.	66 cm, 12.5 cm, 35 cm	9.	11 cm, 12.5 cm, 35 cm
10.	22 cm, 12.5 cm, 0 cm	10.	33 cm, 12.5 cm, 35 cm
11.	44 cm, 12.5 cm, 0 cm	11.	55 cm, 12.5 cm, 35 cm
12.	66 cm, 12.5 cm, 0 cm	12.	77 cm, 12.5 cm, 35 cm
-	-	13.	11 cm, 12.5 cm, 0 cm
-	-	14.	33 cm, 12.5 cm, 0 cm
-	-	15.	55 cm, 12.5 cm, 0 cm
-	-	16.	77 cm, 12.5 cm, 0 cm

4.4.6 Database Enhancement

We offer database enhancement by constructing a synthetic database to tackle sparsity.

4.4.6.1 2D Measurement Database Enhancement

Some data at measured RSSI database or we called as fingerprint are omitted and replaced with artificial data according to Figure 4.15. The blue dots represent the measurement data, and the point other than the blue dots are the points where the data was replaced. From the blue dots, we created the synthetic data. The red dots are reproduced from interpolation or regression using the blue dots on the x-axis. While green dots were created using blue dots on the y-axis. Last step, artificial data in orange dots were made using the green ones.

Later, we order these artificial data as bottom to top rows, left to right columns, e.g., the first red dot in the bottom grid is in the first order (1,0), continue to the second red dot as (4,0). Move to the upper row; the green dot is in order three as (0,1), then continue to order 20 as a red dot in (4,5). The artificial data order and coordinate can be summarized in Table 4.4. In the next step, we added artificial data into the measurement database to have a new database with a 0.5 × 0.5 m2 grid database from

the fingerprint database that we created using bilinear interpolation, polynomial regression, and polynomial interpolation and named as fp+intBil_db, fp+regPoly, and fp+intPoly_db. The method to make the database denser is applied based on one of the authors' works in [102], [158].

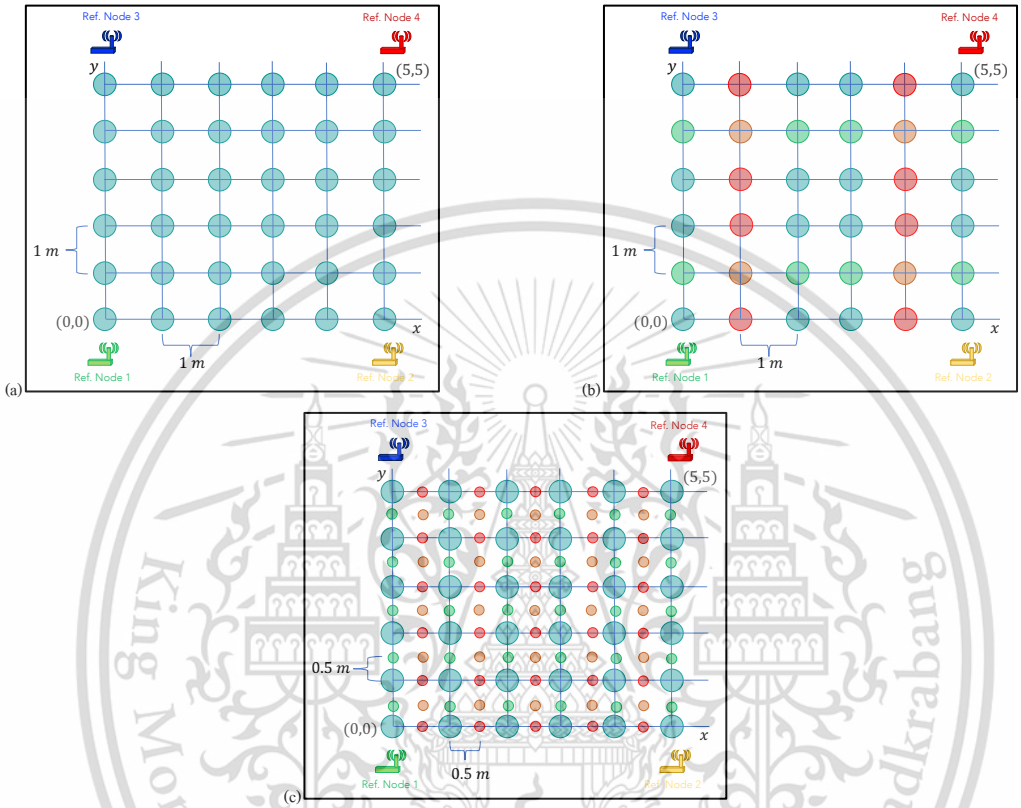


Figure 4.15 Database: a) Actual, b) Replacement scheme in 1 × 1 m^2, c) Synthetic database for 0.5 × 0.5 m^2.

Table 4.4 Data replacement order.

Order No.	Coordinate (x, y, z)		Order No.	Coordinate (x, y, z)	
	x	y		x	y
1.	1	0	11.	1	3
2.	4	0	12.	4	3
3.	0	1	13.	0	4

4.	1	1	14.	1	4
5.	2	1	15.	2	4
6.	3	1	16.	3	4
7.	4	1	17.	4	4
8.	5	1	18.	5	4
9.	1	2	19.	1	5
10.	4	2	20.	4	5

4.4.6.1 3D Measurement Database Enhancement

The 3D measurement database enhancement was applying interpolation and regression to generate artificial data between or within the measured data range; we also created new data outside the range or what we call extrapolation in positions 1, 4, 5, 8, 9, 12, 13, 16 according to Figure 4.13b.

4.5 Results and Discussion

4.5.1 2D Environment

We present the results by using Average Distance Error (ADE) comparison of bilinear interpolation, polynomial interpolation, and polynomial regression for replacement and enhancement data schemes applied as fingerprint databases.

4.5.1.1 RSSI Data Replacement

We examined the discrepancy values between prediction results and the actual RSSI measurements for all reference nodes (four). This RSSI discrepancy is only observed in the replacement order in Table 3. We evaluated the RSSI values predicted results; data is presented in the form of the difference between the data created using polynomial interpolation, polynomial regression, and bilinear interpolation on the RSSI value of the measurement results. Figure 4.16 showed that the highest error value is at order no. 14, with an error from the polynomial interpolation is 5.97 dBm. This result

shows a high difference compared to the error from polynomial regression and bilinear interpolation, with 2.44 dBm and 2.20 dBm, respectively. This result is also the highest among all RSSI prediction error data from reference node 1. While the slightest error is generated at point 11 using the polynomial interpolation method with an error value of 0.00 dBm.

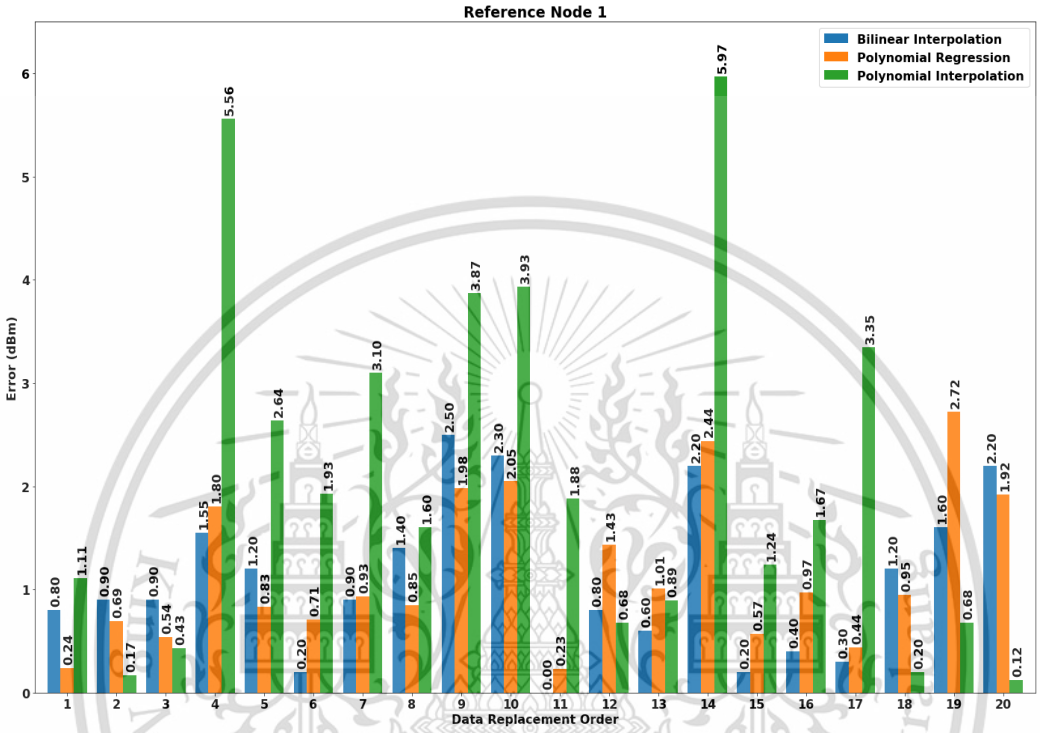


Figure 4.16 RSSI values error for reference node 1.

Figure 4.17 shows the RSSI heatmap comparison for reference node 1. It shows the gradation of the RSSI values as the brighter color refers to the highest RSSI, and the darker color is the low RSSI values. The average RSSI means from the five-time measurement we took the mean value. For interpolation and regression, it was applied to each measurement dataset and took the average values. Figure 16 also shows the similarity pattern between measured RSSI and bilinear interpolation results. On the other hand, polynomial interpolation results are more scattered than the measurement. The results also confirmed the RSSI error discrepancy that is relatively high for polynomial interpolation, as observed in Figure 4.16.

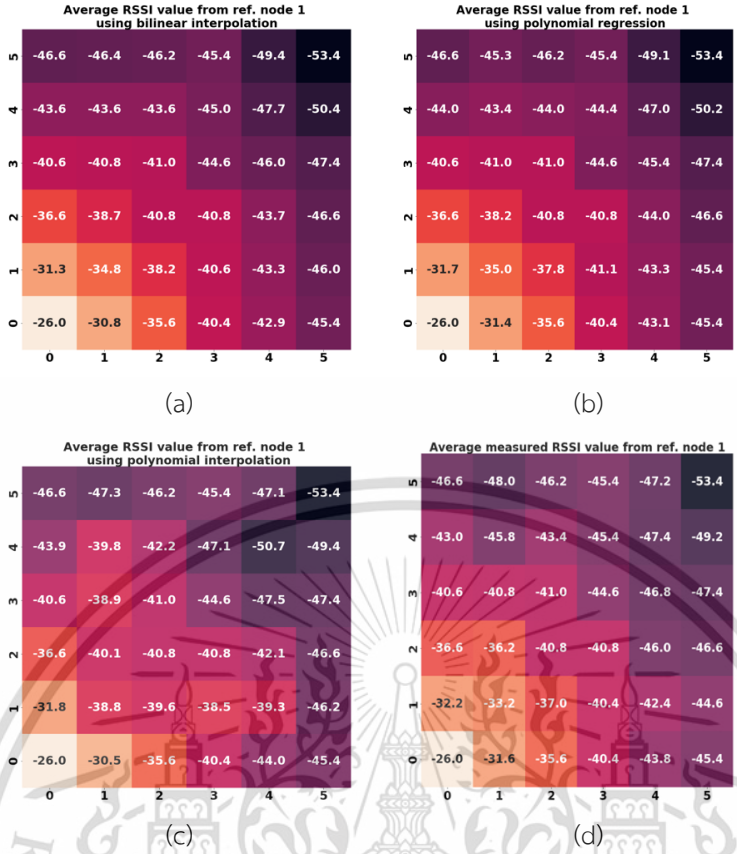


Figure 4.17 RSSI heatmap reference node 1: a) Bilinear inter., b) Polynomial reg., c) Polynomial inter., d) Measurement.

Figure 4.18 shows the RSSI values discrepancy for reference node 2. It shows that the majority of the RSSI values prediction using the polynomial interpolation produces a more significant error than polynomial regression and bilinear interpolation. The most significant prediction errors are at orders 11, 14, and 15, with 3.27 dBm, 3.83 dBm, and 4.44 dBm. On the other hand, the polynomial regression method produces the most significant errors at orders 9, 12, and 19, with each point having errors of 2.38 dBm, 2.73 dBm, and 3.14 dBm, respectively. At the same time, the most negligible error results in the prediction of RSSI values using polynomial interpolation, polynomial regression, and bilinear interpolation are 0.09 dBm, 0.02 dBm, and 0.10 dBm. Figure 18 depicts the RSSI heatmap comparison for reference node 2. The brighter color gradation is origin from the bottom right as the reference node two positions in the measurement.

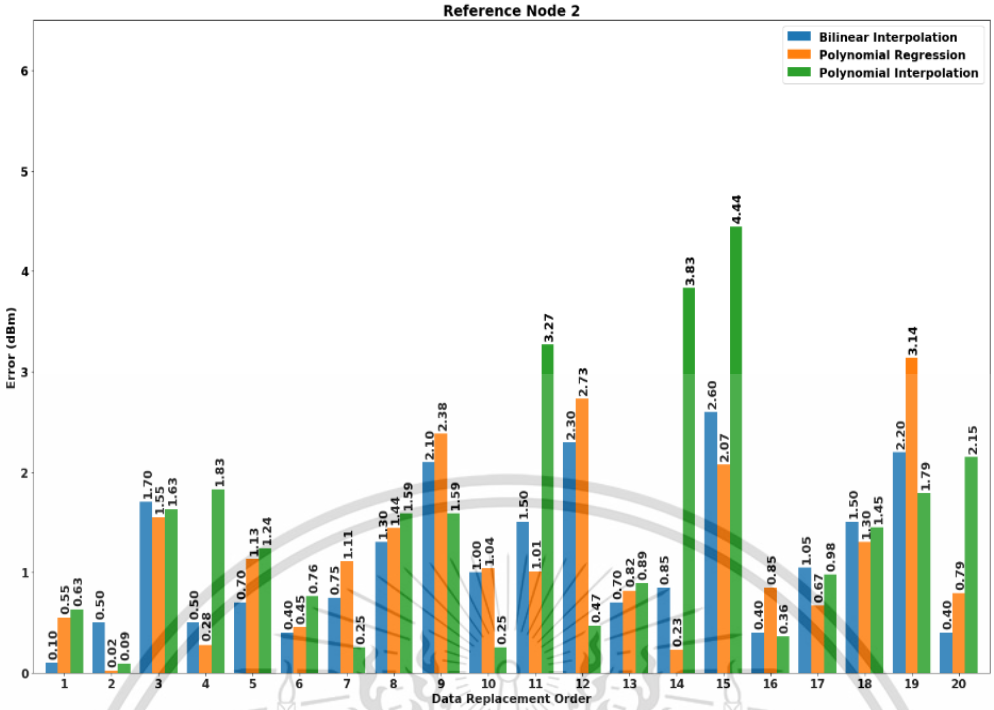


Figure 4.18 RSSI values error for reference node 2.

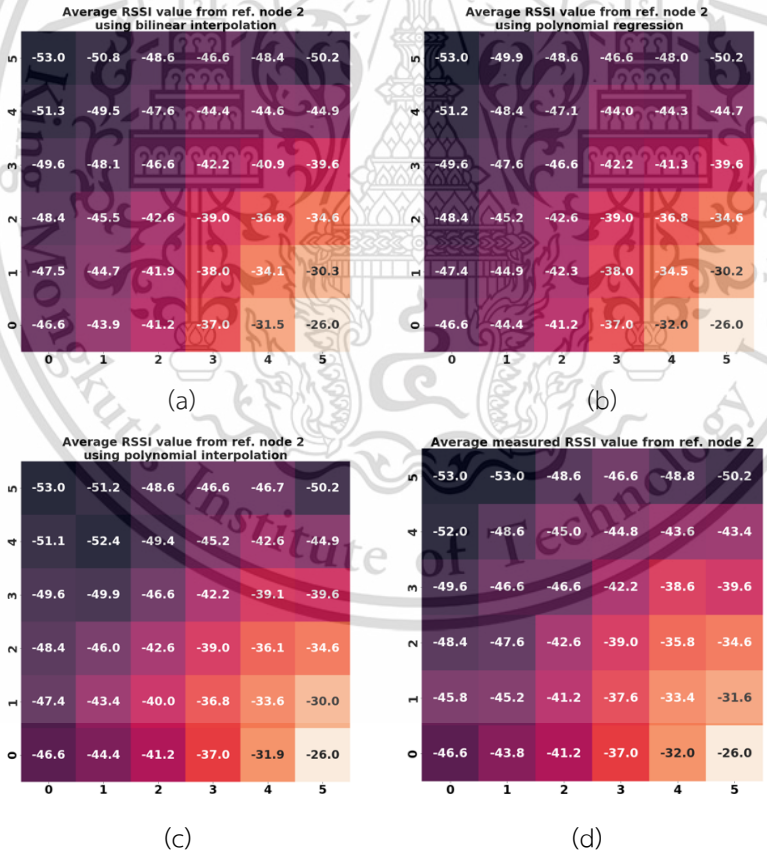


Figure 4.19 RSSI heatmap reference node 2. a) Bilinear inter., b) Polynomial reg., c) Polynomial inter., d) Measurement.

Figure 4.20 shows the RSSI error values discrepancy comparison between the three proposed methods and actual RSSI values from measurement. We found that the most significant errors are at orders 2, 8, and 18, where the replacement of RSSI values using the polynomial interpolation method produces errors of 3.89 dBm, 4.53 dBm, and 3.47 dBm. In reference node 3, the minor RSSI error from polynomial interpolation, polynomial regression, and bilinear interpolation are 0.06 dBm, 0.08 dBm, and 0.10 dBm. Overall, the mean RSSI prediction error using polynomial interpolation, polynomial regression, and bilinear interpolation methods are 1.81 dBm, 0.95 dBm, and 0.81 dBm, respectively. To observe the precise RSSI values distribution, Figure 4.21 shows the RSSI heatmap comparison for reference node three, and the brighter to darker color gradation originates from the upper left (the position of reference node 3 in measurement).

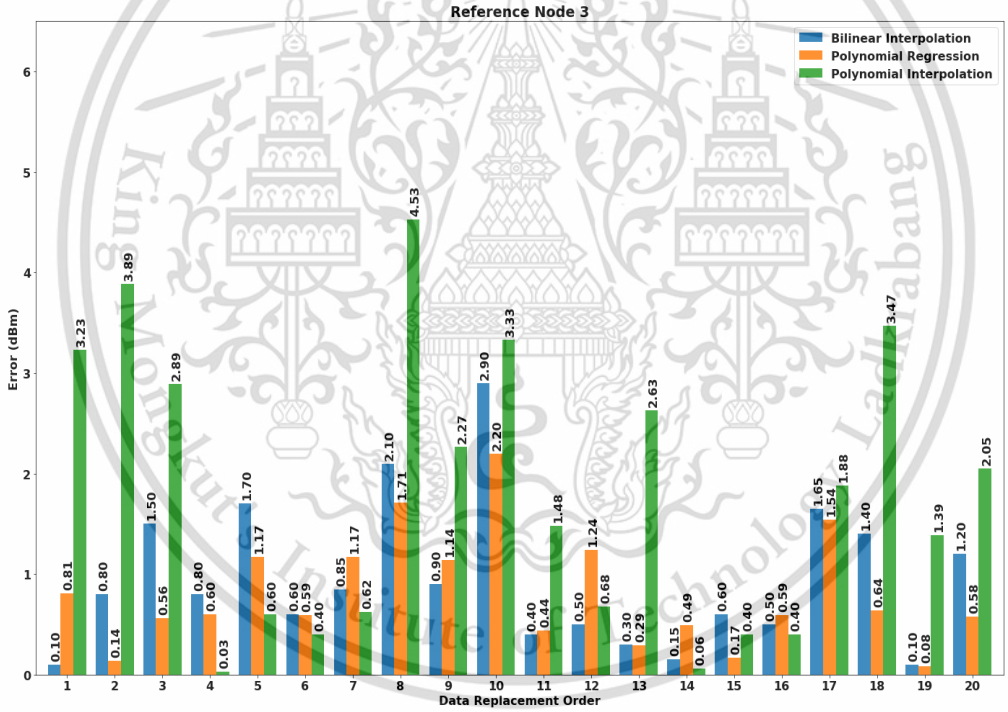


Figure 4.20 RSSI values error for reference node 3.

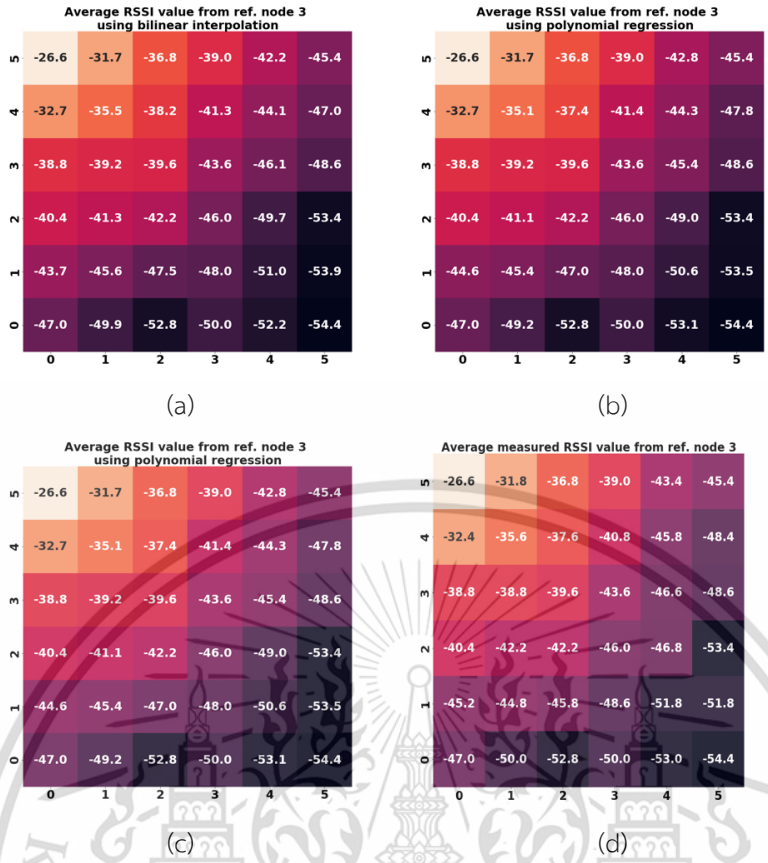


Figure 4.21 RSSI heatmap reference node 3. A) Bilinear inter., b) Polynomial reg., c) Polynomial inter., d) Measurement.

The RSSI values discrepancy from three methods compared to the measurement for the reference node four is shown in Figure 4.22. The results also show that polynomial interpolation produces the most significant error with the largest error at the first order with an error value of 5.11 dBm. In comparison, the polynomial regression method and bilinear interpolation at the same point resulted in an error of 2.92 dBm and 3.90 dBm, respectively. By implementing polynomial interpolation, polynomial regression, and bilinear interpolation methods, the mean RSSI prediction error is 1.74 dBm, 1.16 dBm, and 1.04 dBm, respectively. From these results, we can conclude that the bilinear interpolation performed relatively better than polynomial interpolation and regression. Figure 4.23 shows the RSSI heatmap comparison for reference node 4.

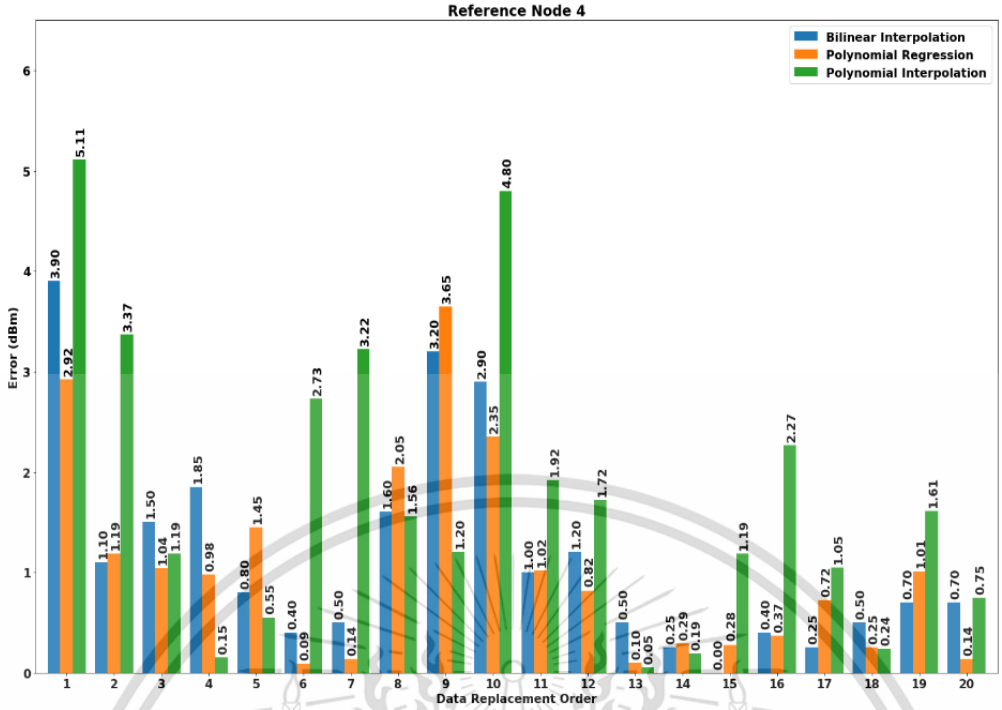


Figure 4.22 RSSI values error for reference node 4.

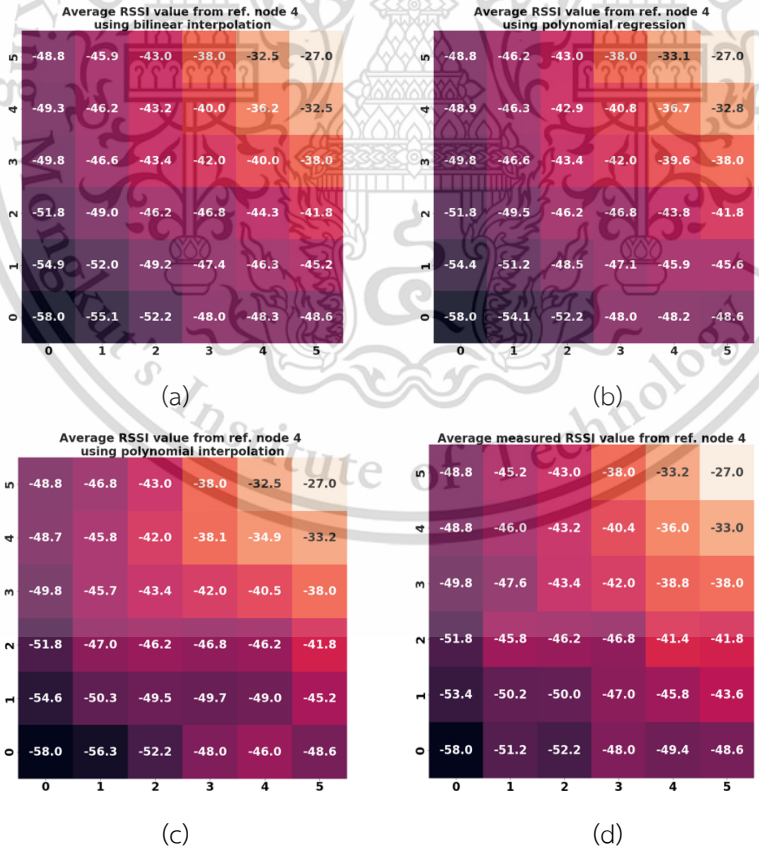


Figure 4.23 RSSI heatmap reference node 4. A) Bilinear inter., b) Polynomial reg., c) Polynomial inter., d) Measurement.

4.5.1.1 Localization Results

The synthetic fingerprint databases are labelled as and fp+intBil_db, fp+regPoly_db, fp+intPoly_db, for bilinear interpolation, regression polynomial, and polynomial interpolation, respectively. For 2D environment, we consider the 4 types of target position namely as target type 1–4. We compare the localization results, fp+intBil_db, fp+regPoly_db, fp+intPoly_db databases with the actual measured fingerprint database, fp_meas_db. For the target types 1-4, the comparison of ADE results can be seen in Figure 4.24. The database used in the positioning results is illustrated in Figure 4.15. By implementing polynomial interpolation, polynomial regression, and bilinear interpolation methods, new data is added so that a database is formed with the amount of data at points where the RSSI value is not measured. The new database is th abelledled as fp+intPoly_db, fp+regPoly_db, and fp+intBil_db. The three databases are then compared with the fp_meas_db database.

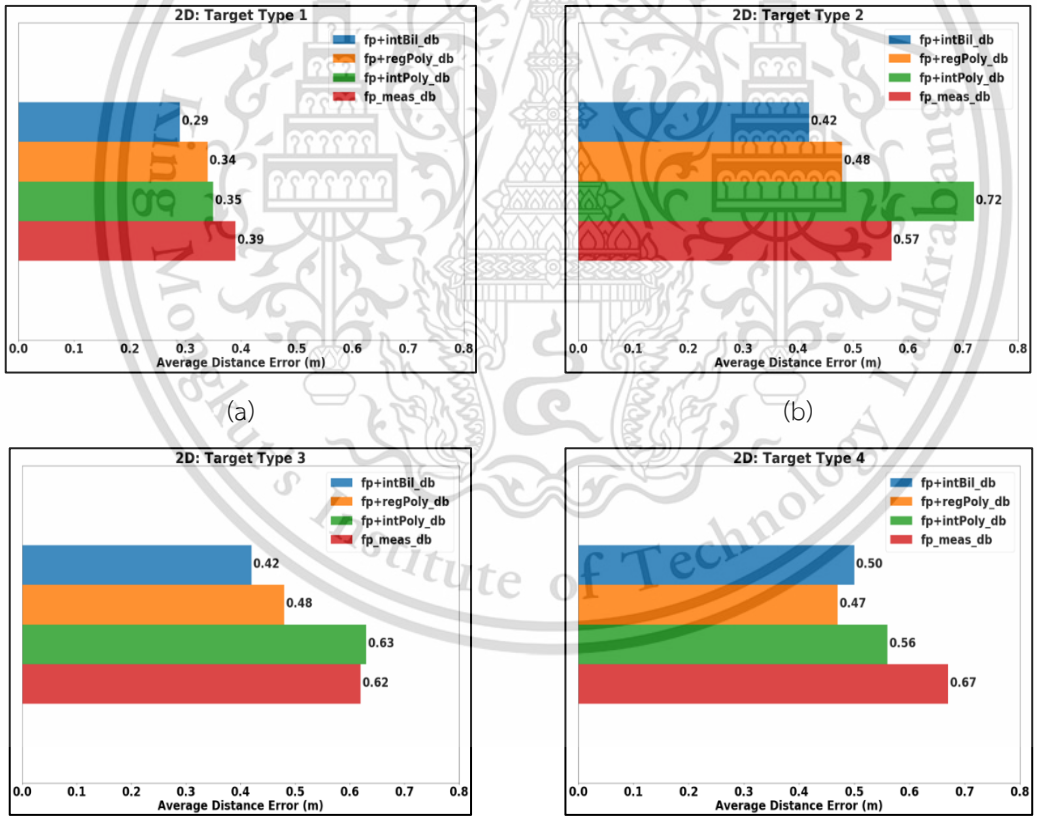


Figure 4.24 Localization results comparison, a) target type 1, b) target type 2, c) target type 3, and d) target type 4.

For target type 1 in Figure 24.a)., the ADE values obtained using fp+intBil_db, fp+regPoly_db, fp+intPoly_db, and fp_meas_db are 0.29 m, 0.34 m, 0.35 m and 0.39, respectively. These results show that applying all the interpolation and regression techniques can reduce the average error of 0.1 m in the target type 1 scenario. The database scarcity, or in other words, by reducing the number of database size in the offline database and we compensated with the classical interpolation and regression technique, has been proven to improve the localization performance.

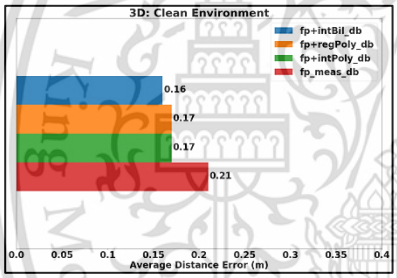
For target type 2, the ADE values obtained using fp+intBil_db, fp+regPoly_db, fp+intPoly_db, and fp_meas_db are 0.42, 0.48, 0.72, and 0.57 m as shown in Figure 24.b). Positioning results with reduced lattice size database using polynomial interpolation method resulted in higher ADE values with an increase of 0.15 m. Meanwhile, polynomial regression and bilinear interpolation reduced ADE by 0.09 and 0.15 m, respectively.

For the horizontal movement of target type 3, the positioning results using fp+intBil_db, fp+regPoly_db, fp+intPoly_db, and fp_meas_db is shown in Figure 24.c). Only fp+intPoly_db obtained the less accurate results compared to fp+regPoly_db, and fp+intBil_db. The 0.2 m error improvement was achieved by the fp+intBil_db, which so far achieved the best results. The last scenario is the target type 4 in the vertical movement. Figure 24.d) shows the results of improvement by all types of the additional database; fp+intBil_db, fp+regPoly_db, and fp+intPoly_db. These results proved that the interpolation and regression for enhancing the database by adding the artificial points between actual measurement points were successfully implemented and validated.

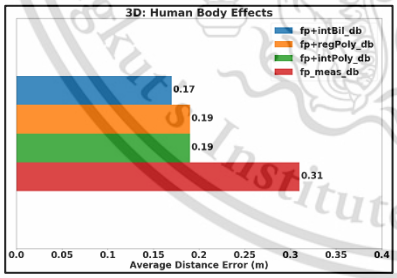
We would like to explore how the RSSI values from 4 reference nodes are placed in each corner of the rectangular shape measurement area by selecting these four target position types. In target types 1 and 2, the diagonal is easier to imagine that the gradation of the RSSI values will follow the diagonal line. However, as the position is somehow in the middle of the area of interest for horizontal and vertical target type positions, it is expected to make the estimation more prone to error, as observed in Figures 24.c and 24.d.

4.5.2 3D Environment

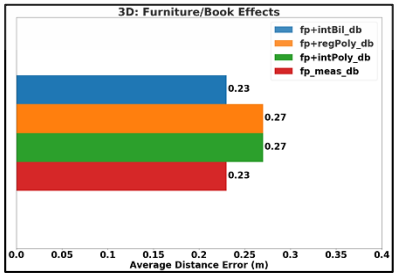
Unlike 2D environment, results for the 3D environment were analyzed and validated by considering three types of scenarios in the bookshelf; the first scenario is a clean environment when there are no obstruction objects on the bookshelf, the second scenario, during the measurement campaign, the first author standing close to the bookshelf (0.5 m) facing front. The third scenario is to place books and other stationaries between the reference nodes in every level of the bookshelf having the reference nodes. We expected that by putting a kind of 'noise' in our measurement, we could observe the improvement when applying the interpolation and regression technique. Fingerprint databases generated from the replacement of measurement data are also labeled as fp+intBil_db, fp+regPoly_db, fp+intPoly_db, and fp_meas_db for bilinear interpolation, polynomial regression, polynomial interpolation, and actual fingerprint database, respectively. These techniques not only augment the data inside the data range but are also used to generate data outside the range (i.e., extrapolation).



a) Clean Environment



b) Human body effects



c) Furniture/books effects

Figure 4.25 Localization results comparison in 3D environment.

Figure 4.25 shows the results for three scenarios in the 3D environment. A clean environment can also be called ideal conditions that we set up without any obstacles in the area of interest. We observe that all enhancement methods can reduce ADE compared to actual fingerprints in this condition. Data quantity enhancement using bilinear interpolation has the most significant ADE reduction. Polynomial interpolation and regression have the same results as shown in Figure 25.a). In the human body effects scenario, we can observe that the localization results were less accurate than the clean environment.

For human body effects, an approach of interpolation applying Kriging for wireless body area network (WBAN) indoor localization has been proposed in [159]. Kriging is a mechanism to predict spatial information for estimating a value at a specific location using actual locations. The paper discussed the possibility of implementing Kriging interpolation to the constructed artificial database and found that by Kriged fingerprint, the accuracy is improved. In this paper, we did not analyze the propagation mechanism. However, we expect that a person standing near the bookshelf will likely affect the localization results. From Figure 25.b), there was a slightly added measured error for 3~8 cm. Polynomial interpolation and regression achieved the same result, but they were less accurate than the actual measurement fingerprint data. Bilinear interpolation has a very tiny accuracy margin better than measurement fingerprint database.

For the furniture effects scenario, all enhancement methods can reduce ADE almost twice than using measurement fingerprint data. In other case studies, positioning results using polynomial interpolation and regression gave the same results in furniture obstacles. The bilinear interpolation always gave the minor error position. The object's existence between reference nodes created shadowing effects that even the synthesis database cannot handle. From these results, we can conclude that in the 3D cases, the signal obstruction in the horizontal direction affected the localization results because of the random signal fluctuation. Table 4.5 compares the results between our proposal and other studies. We emphasized the interpolation technique, the environment, and the performance metric.

Table 4.5 Performance comparison with other studies.

Studies	Parameter and Technology	Techniques	Environment	Performance Metric
The proposed method	RSSI, ZigBee	Bilinear, polynomial, and polynomial regression	Real environment: 2D and 3D	Interpolation reduces estimated error up to 0.2m for 2D and 0.13m for 3D.
Jingxue Bi, et.al. [160]	RSSI, Wi-Fi	Crowdsourcing and interpolation	Real environment: total 3200 m^2 area.	Improving the accuracy results up to 20% manual fingerprint, by using only 25% reference points.
Yanwei Li, et.al. [161]	RSSI, Wi-Fi	Classic narrowband path loss model	Dataset from [162]	Refined map with size 0.75× 0.75 m^2 (grid) improve accuracy up to 63% from 3.6164 m to 1.3294 m

Julie Yixuan Zhu, et.al. [163]	RSSI, Wi-Fi	Spatio-temporal (S-T) similarity model which	Real environment, area of 36×18 m^2	By using the proposed method, the improvement in interpolation and positioning accuracy up to 7% and 32%, respectively.
Yongliang Sun, et.al. [164]	RSSI, Wi-Fi	Fuzzy-C-Means and interpolation	Rectangle area with dimensions 51.6×20.4 m^2	Localization results using kNN fingerprinting yield improvement 0.23m
V. Moghtadaie e, et.al.	RSSI, Wi-Fi	New interpolation method, zone-based Weighted Ring-based (WRB).	2D assumption: 51×18 m^2	Decreases the error (up to 57%) between the measured and reconstructed RSSI values.

The other published studies, mainly Wi-Fi-based RSSI as the Wi-Fi is more popular. However, if considering IoT-based indoor localization, we select the ZigBee standard devices in our case. The techniques and methods mentioned in Table 5 can vary from linear interpolation to path loss model-based RSSI data reconstruction. Most of the results of these publications and our proposed method can reduce the localization errors.

4.6 Conclusion

The database enhancement by applying interpolation and regression techniques to tackle one of the drawbacks of the fingerprint technique is presented. The actual measurement campaign in 2D and 3D environments was conducted, and RSSI values were used for the database fingerprint for both environments. We used a device for measurement based on IoT technology which cost-effective and straightforward implementation of ZigBee standards. The WSNs-based indoor localization is built and validated by some scenarios; the four target type placements, namely diagonal positions both left and right, the horizontal and vertical position for the 2D environment, while three scenarios tested 3D environment; clean, human body effects, and furniture effects. From all scenarios, almost all results agreed that enhancing the database to expand the database grids by artificial data can reduce localization prediction error. From all scenarios, the simple bilinear interpolation stood out as the best accuracy performance. For the 2D environment, it can reduce to 0.2 m error, and 0.13 m for the 3D environment, which is in this environment the grid size is 0.22 m (22 cm). The performance improvement by replacing and adding the synthesis or artificial database can facilitate actual implementation. Thus, the sparsity issues in the fingerprint database can be tackled, and the offline database construction will be less burden. Our research interest is in fingerprint database enhancement; for the next step, we consider applying machine or deep learning-based both supervised and unsupervised learning, both offline and online phase to increase the performance metric of indoor localization system and its flexibility and adaptability in the dynamic indoor environment.

Chapter 5

Database Enhancement via Variational Autoencoders (VAEs)

The intro into deep generative model (DGM) is usually started with the Autoencoder and its improvement, i.e., Variational Autoencoders (VAEs). In this chapter, we discuss about the implementation of VAEs to generate the "fake" RSSI to improve the density and number of RSSI samples.

This chapter is based on our publication [22]:

> Dwi Joko Suroso, Panarat Cherntanomwong and Pithikate Sooraksa, "Deep Generative Model-based RSSI Synthesis for Indoor Localization," 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON), Prachuap Khiri Khan, Thailand, 2022, pp. 1-5, doi: 10.1109/ECTI-CON54298.2022.9795409.

Supplementary data and implementation code in Python can be accessed at https://github.com/dwijokosuroso/VAEs_RSSI

5.1 Abstracts

Indoor localization via deep learning (DL) is attracting researchers' attention. DL is mainly used for fingerprinting-based indoor localization as it generally employs a vast offline database to ensure its reliability. However, the long effort and high cost of constructing this database are the disadvantages of this technique. This paper implements variational autoencoders (VAE), one of the popular deep generative models, to alleviate the drawbacks of offline database issues. Our proposal works using the received signal strength indicator (RSSI); unfortunately, it is known for its fluctuation and instability. Thus, instead of using RSSI directly as a localization parameter, we learn its distribution via VAE to generate the synthetic RSSI values. We utilized the RSSI from an actual measurement campaign. The VAE implementation results show that we can obtain the RSSI synthesis by exploring the latent distribution learned from the input distribution. Thus, the offline database density grids can be enhanced. We validated the results by varying epochs to map the learned latent distribution. However, we still have relatively low accuracy in the synthetic RSSI values, especially when applying a small number of epochs, i.e., 10 and 100. When we applied epoch number 1000, the error was relatively low (-3dBm average error) in the sampled position. Our preliminary assumption is that the dataset is small for VAE learning, and probably the 3-by-3 RSSI-to-image size assumption could still be inadequate.

Keywords: deep learning, indoor localization, fingerprint technique, variational autoencoders, RSSI.

5.2 Introduction and Background

Deep learning (DL) is a popular research topic that has recently been proven to assist in solving many problems in immense fields [1]. One of the hot applications of DL is in the wireless communication field. One branch of research in wireless communication is indoor localization. Indoor localization is still a vibrant research topic because of its usefulness, especially for indoor location-based service (ILBS) [2].

The general techniques of indoor localization are range-based and range-free. This range-based term means localization parameter translation to a distance with the other process to an algorithm estimating the target/object location. On the other hand,

the range-free technique employs "fingerprint," which, instead of using distance translation from a parameter, utilizes the parameter's spatial information in specific locations and stored as the offline database. As this technique has a two-phase process, the offline database construction and the online phase as the location estimation phase, this technique's major drawback is the effort and high cost of constructing the offline database [3].

This paper proposes a solution to alleviate the fingerprint-based technique drawback in the offline database by generating the synthetic received signal strength indicator (RSSI) via generative DL. RSSI is a most specific parameter which relatively simple and available for extraction from Wi-Fi-based devices. In this paper, we implemented one of the famous deep generative models, the variational autoencoders (VAE)[4].

Similar research has been published, i.e., authors in [5], [6] applied channel state information (CSI) for VAE input. However, acquiring CSI is quite complex and needs high cost and sophisticated equipment for its extraction. Another publication proposes the semi-supervised VAE in indoor localization using RSSI [7]. The other researchers apply the generative adversarial networks (GAN) [8] to augment RSSI values and localize the target [9], [10]. GAN is more reliable with the vast dataset; however, it costs the training time since it is based on adversarial networks which employs two dueling neural networks.
Most of these current publications in the deep generative-based model for database enhancement consider the available measurement dataset, a relatively high number, and dense datasets. In our approach, we employed the RSSI values from an actual measurement campaign which was more straightforward and had relatively low density. We utilized a Wi-Fi-based low-cost device, ESP32, and considered 8 (eight) reference nodes (RNs) in a 5-by-5 m2 area of interest in a typical classroom with 25 fingerprint locations in 1-by-1 m2 grids.

Our preliminary results can increase the number of databases by RSSI synthesis via VAE's latent distribution. By choosing the latent distribution's specific points, we can even obtain the RSSI values (synthetic) in the area outside the 25 assigned fingerprint locations. In terms of the dataset used, we found that our number of datasets is still relatively low to train VAE well, so the results of RSSI synthesis still need some improvements.

This paper is structured as follows; the paper is introduced in the first part. We discuss materials and methods in the second part. The measurement campaign and

research flow are presented in the third part. We present the results and discussion in the fifth part before we conclude our findings and discuss our near future works in the last part of the paper.

5.3 Materials and Methods

5.3.1 RSSI and Fingerprint Technique

RSSI is the signal strength-based parameter indicating the received power by a receiver. We have published the direct use of RSSI for indoor localization parameters applying machine learning and interpolation techniques in [11], [12]. RSSI values fluctuate with time and multipath effects of a complex indoor environment. It is essential to learn the RSSI distribution instead of directly using it as the localization parameter. One of the most used techniques in indoor localization is the fingerprint-based technique [13]. Figure 1 shows the fingerprint technique illustration showing a two-phase process, offline and online. The area of interest is designed in such a way to accommodate the fingerprint grids to collect localization parameters, i.e., RSSI in dBm. In this example, the parameter is stored from 3 (three) reference nodes for all fingerprint locations as the fingerprint database in the offline phase. The online localization phase happens when the target or object sends the new RSSI and compares it with those in the database by the pattern matching algorithm to infer the target's location by the similarity of their RSSI values.

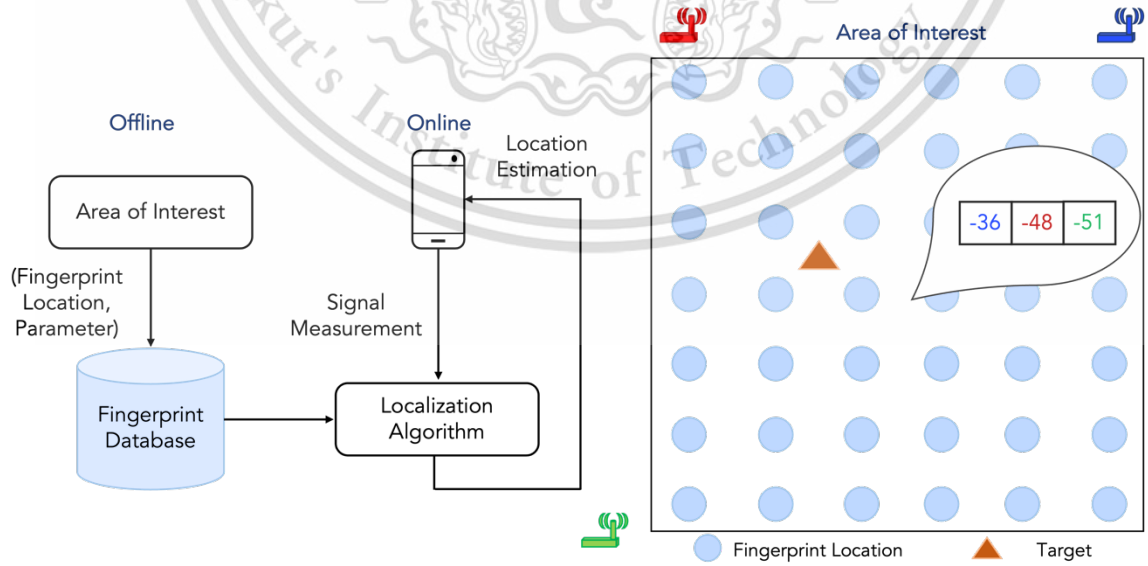


Figure 5.1 Fingerprint technique illustration.

5.3.2 Variational Autoencoders (VAE)

The basic understanding of the generative model is to learn the input and generate similar values as the output by the latent distribution learned from the input distribution. One of the generative DL models is variational autoencoders (VAE). VAE is the improved version of traditional autoencoders based on variational interference proposed by [69]. To describe the VAE, we can begin by quantifying information and definition of entropy. Given information as I, and $p(x)$ as the probability of an event x, such as

$$I = -\log p(x) \tag{5.1}$$

When $p(x) = 1$, the highest probability, there will be no information; contrary, it will give more information when there is a low probability. While the entropy, H, is the expectation value of the information shown as Equation (5.2).

$$H = -\sum p(x) \log p(x) \tag{5.2}$$

In VAE, the output or reconstructed part needs the latent distribution, z. In real world, the information is limited on output, $\hat{x}$. Thus, it is more reliable to infer the z via the posterior distribution, $p(z|x)$, rather than via $p(z|\hat{x})$. As the graphical model, the VAE's obstacle is that the $p(z|x)$ is intractable.

$$p(z|x) = \frac{p(x|Z)p(z)}{p(x)} = \frac{p(x,z)}{p(x)} \tag{5.3}$$

We can approximate the $p(z|x)$ via variational inference. Let $q(z)$ represents the prior distribution of z. Then, we need the Kullback-Leibler (KL) divergence to estimate the distribution similarity of $p(z|x)$ and $q(z)$. The minimum value of KL refers that these two distributions are similar.

$$\min KL(q(z)\|p(z|x)) = -\sum q(z) \log \frac{p(z|x)}{q(z)} \tag{5.4}$$

By deriving the Eq. (4)'s right-side term,

$$
\begin{aligned}
-\sum q(z)\log\frac{p(z|x)}{q(z)} &= -\sum q(z)\log\frac{p(x,z)}{q(z)}\cdot\frac{1}{p(x)} \\
&= -\sum q(z)\left[\log\frac{p(x,z)}{q(z)}+\log\frac{1}{p(x)}\right] \\
&= -\sum q(z)\left[\log\frac{p(x,z)}{q(z)}-\log p(x)\right] \\
&= -\sum q(z)\log\frac{p(x,z)}{q(z)}+\sum_z q(z)\log p(x) \\
&= -\sum q(z)\log\frac{p(x,z)}{q(z)}+\log p(x)\sum_z q(z)
\end{aligned}
\tag{5.5}
$$

With $\sum_z q(z)=1$, we then obtain $\log p(x)$ by arranging Equation (5.5) to (5.4).

$$
\log p(x) = KL(q(z)\|p(z|x)) + \sum q(z)\log\frac{p(x,z)}{q(z)} \tag{5.6}
$$

The first term in Eq. (6) is KL divergence which must be minimized to make two distributions similar, and the second term is the evidence lower bound (ELBO) that needs to be maximized. The second term can be further expressed by,

$$
\begin{aligned}
-\sum q(z)\log\frac{p(x,z)}{q(z)} &= \sum q(z)\log\frac{p(x|z)p(x)}{q(z)} \\
&= \sum q(z)\left[\log p(x|z)+\log\frac{p(x)}{q(z)}\right] \\
&= \sum q(z)\log p(x|z)+\sum q(z)\log\frac{p(x)}{q(z)}
\end{aligned}
\tag{5.7}
$$

The first and second term in Eq. (7) expresses the expectation values w.r.t distribution of $q(z)$ and KL divergence of $-KL(p(x)\|q(z))$, respectively. The variational inference of this process is illustrated in Figure 5.2.

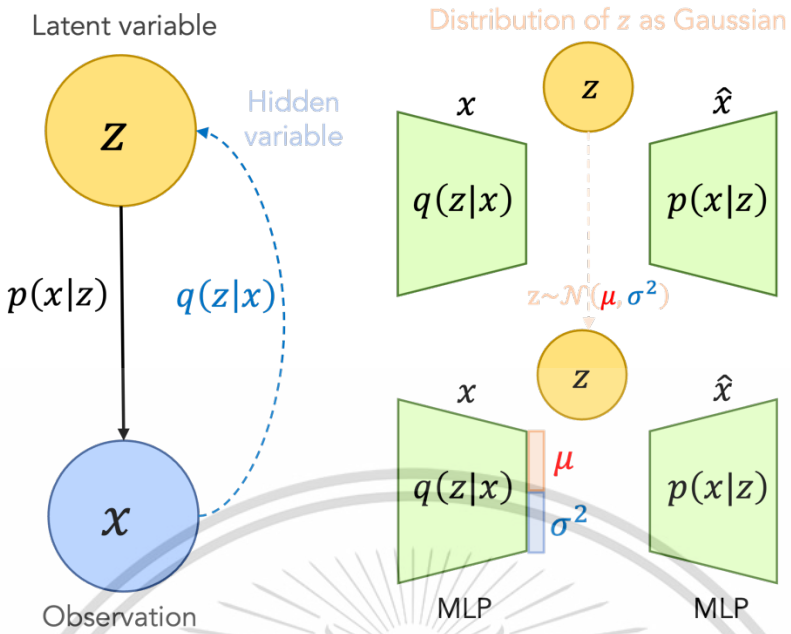


Figure 5.2 Variational inference.

As $p(x|z)$ is intractable, we can instead estimate the $q(z|x)$ by the KL divergence concept to find $p(x|z)$ by analogizing the distribution of $q(z|x)$ following Gaussian distribution with mean and variance, μ and σ^2. The μ and σ^2 are the outputs from the encoder part by multilayer perceptron (MLP). The illustration of this VAE process is depicted in Figure 5.3 [81].

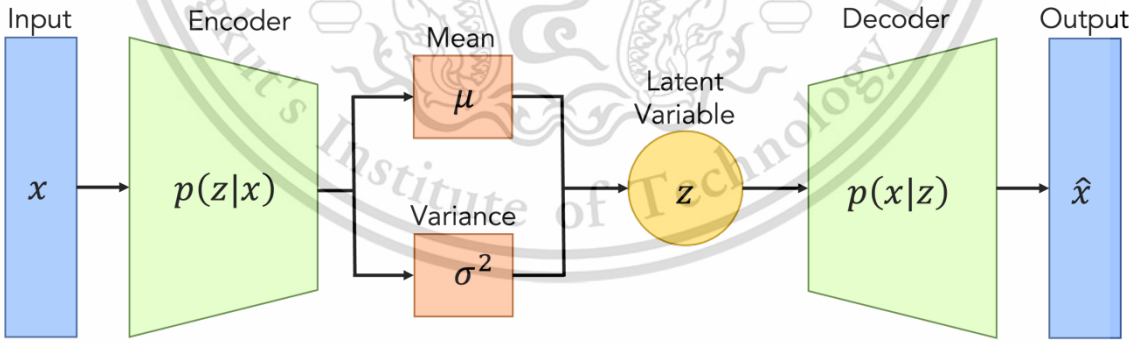


Figure 5.3 VAE illustration.

The Gaussian distribution is assumed in the case of real-valued data as the data is generated by some random processes which involve an unobserved continuous random variable. These random processes, e.g., measurement and data distribution, generally will follow the Gaussian/normal distribution. Unlike the traditional autoencoder directly constructed via input, the VAE generates the output via the latent

distribution, z by given input, x as $z \sim \mathcal{N}(\mu, \sigma^2)$. Thus, the objective function, $\mathcal{J}$ is expressed by Equation (5.8).

$$\mathcal{J} = \min_{0} \|x - \hat{x}\|^2 + KL(q(z|x)\|\mathcal{N}(\mu, \sigma^2)) \quad (5.8)$$

The first and second terms represent the reconstruction and KL loss, respectively. We now have the mean and variance sampling from the distribution of z and further feeding it to the decoder. Since all x's have been mapped to z with its distribution, we can generate the data/points like the input. This paper proposes the VAE to generate the synthetic RSSI given by the RSSI obtained from the actual measurement campaign as x. The output is the synthetic RSSI, $\hat{x}$, which is related to some latent variable, z.

5.3.3 Measurement Campaign

5.3.3.1 Measurement System and Setup

The size of the classroom area for the measurement campaign is $10 \times 15.14\ m^2$. We considered the 25 fingerprint locations in a $5 \times 5\ m^2$ area of interest and recorded the RSSI values from 8 reference nodes (RNs). The total dataset for these locations is 1,489 rows, and 8 (eight) columns represent the number of RNs. The data were collected by the 3 minutes of transmit-receive RSSI values for all 25 fingerprint locations, collected sequentially for each location. Each fingerprint location has around 60 sets/rows of RSSI values from these 8 RNs. We considered this span duration to limit the time-varying effects on our RSSI values. The measurement illustration layout can be depicted in Figure 5.4. Figure 5.5 shows the actual measurement location with some interference objects (IOs), e.g., chairs and tables outside the area of interest.

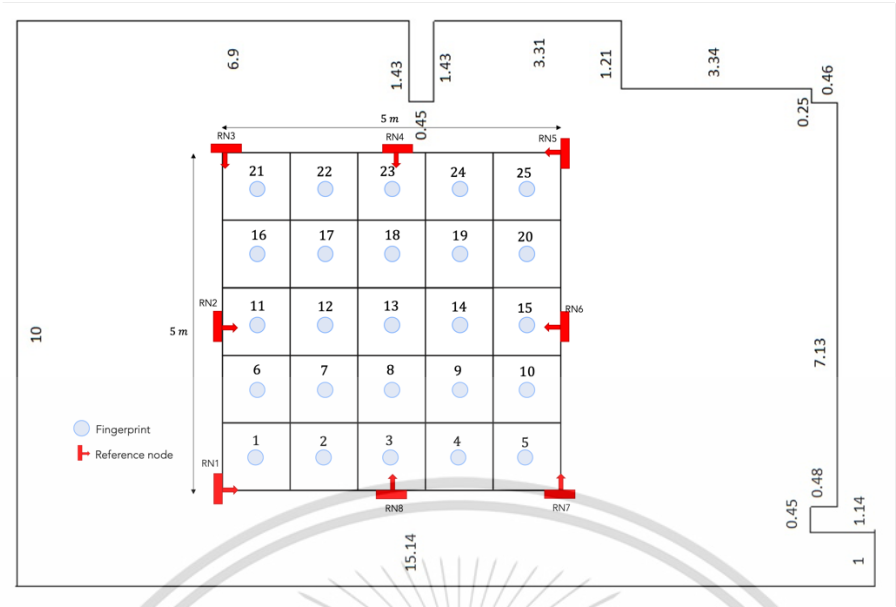


Figure 5.4 Measurement layout illustration.

Figure 5.5 Actual measurement layout.

Figure 5.6 shows the actual ESP32 device installment and illustrates the ESP32 system arrangement; 8 RNs, one as the target, and one server node connected to a computer for RSSI data acquisition. In total, we employed 10 ESP32 units for RSSI database construction.

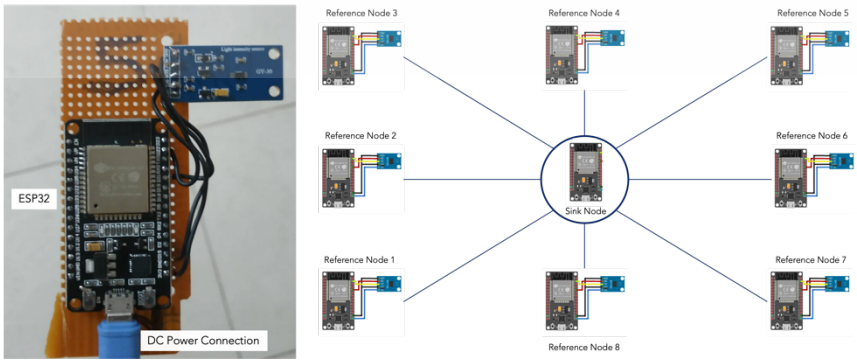


Figure 5.6 Device installment and arrangement.

5.3.3.2 VAE Implementation for RSSI Synthesis

We implement the VAE via the Keras library of Python. Inspired by the image processing approach, we can get different image results by selecting different points in latent distribution obtained from variational inference by input image distribution. The variable of latent spaces is the mean and variance learned from the input distribution. Our proposal utilized the RSSI as a training parameter to generate other RSSI values.

1. ***RSSI as an image:*** Before VAE implementation, we represented RSSI in a 3-by-3 image [165]. The illustration and assigned RSSI values are shown in Figure 5.7. For instance, RSSI_1 indicates the RSSI values From RN 1, and each different values from RNs yield different color. The normalization is done by highest value (100 dBm) and lowest value (-100 dBm) as reference for 0 (min value) and 255 (max value) for grayscale conversion. The null value in the last pixel in our represented image is set to 100 dBm (yellow).

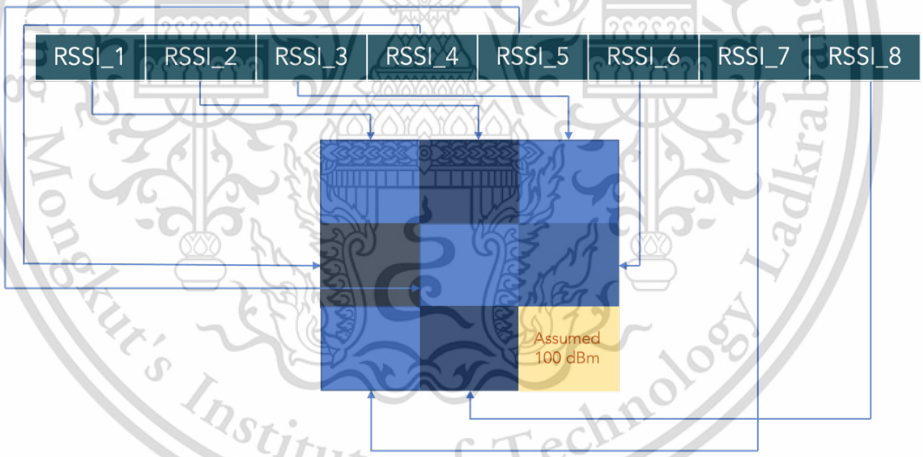


Figure 5.7 RSSI-to-image conversion.

2. ***VAE implementation in Keras:*** The input is RSSI values; we set latent dimension as 2 (two), as it is enough to cover the 3-by-3-pixel RSSI-image representation. We have an input layer for the encoder and four consecutive 2D convolution layers (32 kernels for the first layer, 64 kernels for the rest, ReLu activation). For the decoder, the input size is 2 (two latent dimension assumption), a 2D convolution with the transpose feature (64 kernels, ReLu activation) as in decoder it will proceed from the larger size input from the encoder to smaller size for its output (reshape) to emulate the original input

size. In the VAE training part, we have four 2D convolution layers similar to the encoder part and applied the Adam optimizer. One custom layer is applied to make the consistent size between input in the encoder and output at the decoder.

5.4 Results and Discussion

5.4.1 Latent Distribution vs. Epoch

To accurately synthesize RSSI, we investigated the different number of epochs for the latent distribution results. Figure 5.8 to 5.10 show the latent distribution for 10, 100, and 1000 epochs. The epoch affects the distance between latent space of coordinates sample locations of 1 to 5. In a sense, value number 1 represents the coordinate (1,1) and until five as the coordinate of (5,5). Figure 5.8 shows that the latent distribution of 1-5 is tight and piling up each other. Interestingly, when the number of epochs increases, the clusters and correlation between them are reduced, yielding a different latent distribution. In which we expected a higher accuracy for RSSI synthesis. If the latent distribution has high accuracy in predicting the input distribution, we can get RSSI synthesis accurately.

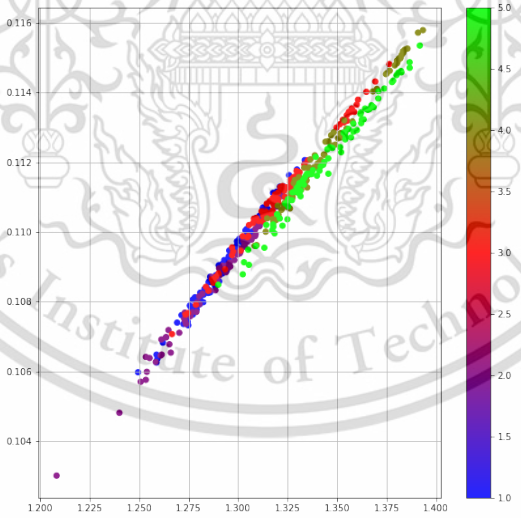


Figure 5.8 Latent distribution vs. 10 epochs.

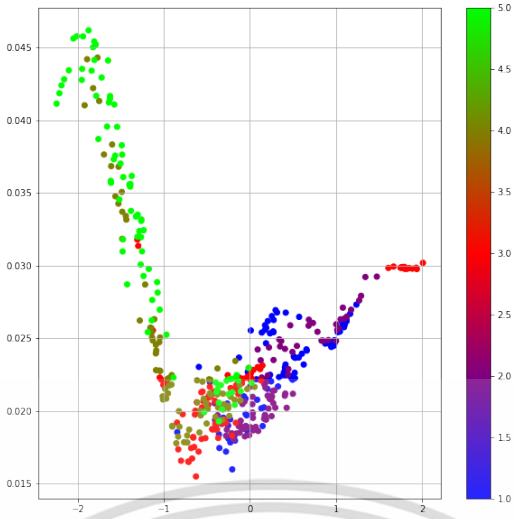


Figure 5.9 Latent distribution vs. 100 epochs.

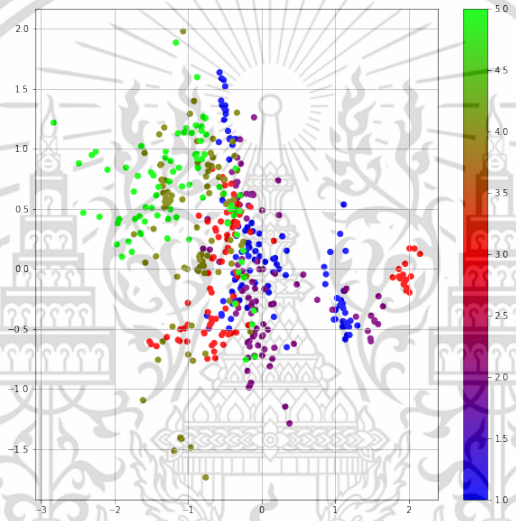


Figure 5.10 Latent distribution vs. 1000 epochs.

5.4.2 Actual vs. Synthetic RSSI

Figure 5.11 shows the RSSI values discrepancy between actual and synthetic w.r.t number of epochs.

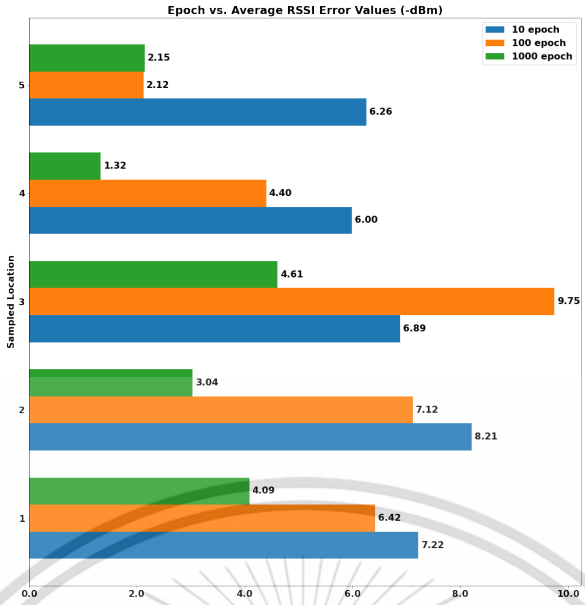


Figure 5.11 Actual vs. synthetic RSSI average error.

We only consider 5 sampled location which is (1,1), (2,2), (3,3), (4,4), and (5,5) as 1 to 5 number in color-bar representation. As our expectation, when we use a higher number of epochs in the learning phase of VAE, we can improve all average discrepancy errors between actual and RSSI synthesis RSSI from the sampled locations. RSSI values discrepancy are affected by the small gaps between latent space representing the RSSI-coordinate values. As shown in Figure 5.8 and some points in Figure 5.9 and 5.10, the latent distributions were piling up; the RSSI values are relatively similar in certain distributions. At the same time, the actual RSSI values are different, especially when we observed RSSI values corresponding to the RN sequential and position. The RSSI visualization for comparison is shown in Figure 5.12. It shows that some discrepancy occurs and varies in RSSI from each RN.

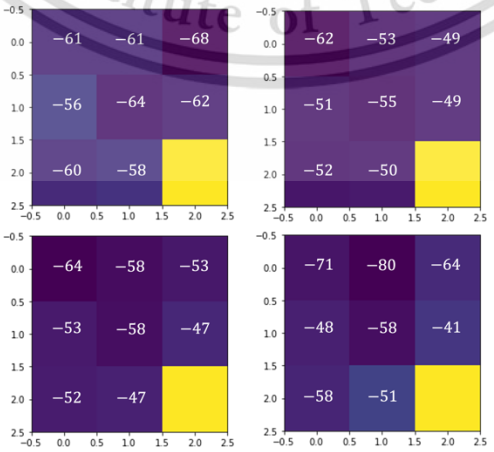


Figure 5.12 RSSI values, clockwise direction: actual, 10, 100, 1000 epochs in (1,1).

5.4.3 Synthetic RSSI Accuracy Discussion

Results in **5.4.2** show that the accuracy is acceptable, especially for 1000 epoch results since the error from 5 positions are relatively low (-3 dBm average error). We want to discuss the preliminary assumption related to these results:

1. ***Sample RSSI for dataset***: We have only 1,489 samples of 8 (eight) RSSI values from the 8 (eight) RNs. As each fingerprint location only has 60 rows, 8 columns dataset, we found that this relatively small dataset would affect the proposed system's learning process.
2. ***3-by-3 image assumption***: We consider reshaping our RSSI values into the 3-by-3-pixel image as an "image" input for VAE. This small pixel size needs further investigation to determine whether it can be the source of some inaccuracy in RSSI synthesis results.
3. ***The number of epochs***: We compared the number of epochs to see whether the number of epochs will reduce the loss function. In our results, the loss and validation loss values are 0.5949, 0.5927, and 0.5902 for 10, 100, and 1000, respectively, as Equation (5.8) suggests that these two losses should be approached 0 to ensure that the VAE training is optimum. However, from Figure 5.11, we can see an improvement in RSSI error values when the number of epochs increases.

5.5 Conclusion

We proposed VAE implementation to synthesize RSSI values from actual RSSI from a relatively small measurement campaign. The results show that the latent distribution in our VAE implementation can be used for RSSI synthesis. Observing the latent distribution from all epochs shows that each cluster of coordinates is not so distinct from other clusters, making some RSSI syntheses inaccurate. The number of RSSI datasets we have, and the small image size assumption (3-by-3) could affect this high correlation between clusters. Our near-future work is to explore denser datasets via both publicly available datasets and conducting measurement campaigns. We will improve the RSSI-to-image conversion assumption, including its size and other considerations. We will also compare VAE results with other deep generative models, e.g., GAN. Then, the combination of actual and RSSI synthesis will also be validated to infer the target's location via the deterministic deep learning method.

Chapter 6

Database Enhancement via Generative Adversarial Networks (GANs)

GANs are one of the most famous deep generative models considered for our fingerprint database enhancement. We base this chapter on our two publications [166]:

1. Dwi Joko Suroso, Panarat Cherntanomwong, Pitikhate Sooraksa, "Fingerprint-based Indoor Localization via Deep Learning ," The 6th International Conference on Electronics, Communications and Control Engineering (ICECC) 2023, Fukuoka, Japan, 24-26 March 2023, ACM Publisher. (*to be published*)
2. Dwi Joko Suroso, Panarat Cherntanomwong, Pitikhate Sooraksa, "Synthesis of a Small Fingerprint Database through a Deep Generative Model for Indoor Localisation," Elektronika ir elektrotechnika, Vol. 29 No. 1, pp 69-75, February 2023. https://doi.org/10.5755/j02.eie.31905

Supplementary data and implementation code in Python can be accessed here: https://github.com/dwijokosuroso/small_GANs

6.1 Abstract

Deep learning (DL) application is proven helpful in a vast research field. The deep generative model of DL is helpful for data augmentation objectives to tackle the lack of datasets that have a significant impact on learning performance. Data augmentation or synthesis is expected to solve the issue in a small/sparse database. One recent trend is to employ deep generative models in radio frequency (RF)-based fingerprint technique for indoor localization as it can solve the problem of its databasing issues. The fingerprint technique is the most used indoor localization technique known for its accuracy and performance. Whilst the fingerprint technique performance is solely depended on the database density, the dense offline fingerprint database must be constructed with the accuracy requirement. However, this will affect the high cost, massive laborious work, and increase the complexity of the system. Moreover, to apply deep learning, we also need a large dataset for it to learn efficiently. We propose to implement a DL-based fingerprint technique to tackle both problems of dataset scarcity by generating synthetic data via a deep generative model and increasing localization performance. We propose the DL's discriminative model, i.e., multilayer perceptron (MLP), for classification tasks. For the fingerprint database augmentation, we employed the generative model, i.e., Generative adversarial networks (GANs). We considered using a received signal strength indicator (RSSI) from a measurement campaign based on Wi-Fi devices for the database. The total area of interest for measurement is 25 m^2 inside the typical classroom environment, and we consider the 25 fingerprint locations as labels. We have a dataset of 1,250 rows x 8 columns (from 8 reference points). From the results, by using only 50% of actual data combined with the 125 synthetic data, we can improve the accuracy by more than 200% compared to only using 50% of actual data and show a 60% improvement in the loss. The combination of 100% actual data and 125 synthetic data gives the best accuracy and loss performance of 0.76 and 0.85, respectively. It gives an improvement of 144% in accuracy and 200% loss performance. By implementing deep learning for fingerprint techniques for data augmentation and classification, we can achieve good performance and reduce the workload of fingerprint database construction by data synthesis through GANs by using only a small dataset.

6.2 Introduction and Background

Indoor localization is still an active research topic today. The most widely used indoor localization techniques are distance-based and distance-free. Distance-based refers to the technique when the parameter needs to be converted to a distance and use a specific method to find the location, e.g., trilateration, angle-based, and time-based methods. However, this technique is reliable only when achieving good quality from the parameter-distance conversion result. The complex indoor environment yields a multipath effect in radio propagation; the non-line-of-sight condition often degrades the quality of the parameter [2], [16].

On the other hand, the distance-free method, i.e., the fingerprint method, does not need the parameter-distance conversion. It has benefit on using spatial information directly which consist of the location of the fingerprint points and their corresponding localisation parameter values. The multipath effect in the indoor environment does not affect the localization process, because all these effects are considered in the recorded database [167].

The fingerprint technique has been a central and exciting topic in indoor localization. Despite its accuracy, performance, and simplicity of parameters, the fingerprint technique has a significant drawback in its offline database construction. This database needs to be densely designed to ensure the reliability of accuracy performance. However, much laborious and cost efforts are needed to build a very dense database. Instead of collecting the dense database in the offline phase, this paper tackles this problem. The small number or scarce location of fingerprints is emphasized. Then, data synthesis is applied to make the database denser [168].

Previous publications offered some approaches to the fingerprint issue in offline fingerprint construction, including the crowd-sourcing data; unfortunately, active participants are needed in the data collection process. The basic idea of our proposal is how we can generate synthesis/fake database to compensate for the low density of the fingerprint database. We apply deep generative model-based data synthesis as deep learning (DL) implementations have been widely proven to be helpful in a wide area of research [14], [169].

As DL-based research have had tremendous attention in the last decade. The support of computational power, the research in advanced computer code, and the broad access availability of DL courses and education open its potential for a broad

research field [63]. One recent trend in wireless communication, especially in radio frequency (RF)-based indoor localization, is applying deep learning for data augmentation and classification tasks [170]. Data augmentation is essential in training efficiently because DL needs a large amount of data. In the same viewpoint, one of the popular indoor localization techniques, i.e., fingerprint technique, also needs a high amount of fingerprint database number or density to fulfill accuracy requirements [77], [170]. The DL implementation can tackle data scarcity and improve localization performance [171].

DL consists of two primary models: discriminative and generative. The generative model is not popular initially because of the difficulty in the posterior distribution estimation. For example, it is found in the explicit density-based model, i.e., variational autoencoders (VAEs) [63], [66]. Ian Goodfellow et al. introduced generative adversarial nets (GANs) in 2014. The authors made a breakthrough finding in the deep generative model by dueling two deep neural networks instead of estimating the intractable probability distribution [172]. The DL's discriminative models include the multilayer perceptron (MLP), the convolutional neural networks (CNNs), and Long-short Term Memory (LSTM). The discriminative models are mainly used for the classification task [14], [66], [173].

From the fingerprint technique perspective, the sole reason for high localization performance is to make a high-density and high-quality database [18]. However, to do so, the pain point of this technique arises. Extensive measurement campaigns to collect the database are needed to ensure high database density. Moreover, the cost and labor are pervasive when the indoor environment is large-scale and involves many devices. These drawbacks hinder the implementation of a fingerprint-based technique for indoor localization [19]. Some published works focused on the enhancement of the multifeatured fingerprint database, e.g., by combining the received signal strength indicator (RSSI) and illumination [24], RSSI and time difference of arrival (TDOA) technique [174], and modified fingerprint to work for heterogeneous network and optimize k-nearest neighbor (kNN) algorithm for localization [175].

The number of databases is enhanced through crowdsourcing and database synthesis or augmentation [77], [85]. We will discuss more in the latter enhancement. Data augmentation using a generative model is proposed to tackle the issue of database scarcity. Authors have published on this topic, e.g., the preliminary RSSI synthesis by VAEs implementation [22]. Some recent works in the GANs for data augmentation have been

published in [76], [77]. While for target classification or pattern matching algorithms, CNNs can give high performance [176], [177].

We offer a lightweight and straightforward implementation of DL for fingerprint techniques. We proposed both discriminative and generative in our localization system. Like [76], we propose to augment the fingerprint database from the offline phase, in this case, RSSI synthesis using GANs. In the online or pattern-matching phase, we employ the feed-forward neural network, the MLP. The difference is that instead of using the rooms as labels, we use labels for different fingerprint locations since our database is relatively small. The synthetic data via GANs is flexible since it is unsupervised work and does not need the labeled data, which we know are expensive. Data augmentation aims to assist and improve classification training and validation.

We evaluate our proposal by having a measurement campaign in the typical indoor classroom of 5m x 5m with the Wi-Fi-based device ESP32 and applying the star topology of wireless sensor networks (WSNs). The reference points (RPs) are 8 in total, and the total RSSI dataset is 1,250 rows x 8 columns (representing the number of RPs). The system validation takes incremental from 10% to 100% of actual chunk data and directs classification via MLP. At the same time, the chunk data is used to generate RSSI synthesis via GANs. At last, the MLP is also used to classify the RSSI data for the combination of actual and synthetic data.

Our emphasis in this study is:

1. implementation of DL for the offline phase (data augmentation) and online phase (classification).
2. the RSSI is from an actual measurement campaign and is relatively small.
3. we compare the actual and the combination of actual and synthetic data for the localization performance. We also consider two additional number of synthetic data: 125 and 625 and compared their performances.

6.3 Materials and Methods

This part presents the concept of indoor localization and fingerprint technique. An explanation of deep learning's discriminative and generative models will be presented.

6.3.1 Indoor Localization and Fingerprint Technique

An indoor localization system is defined as a system to infer the location of the object(s) or target(s) in an indoor environment [16], [19]. Different from the established global positioning system (GPS), which works reliably in the outdoor environment, indoor localization utilizes some technologies, e.g., radio frequency (RF), optical sensors, acoustic, and inertial measurement unit (IMU). From the ease of implementation, RF-based, i.e., Wi-Fi-based indoor localization, has the most significant future applications [15].

The multitude of prospective services enabled and applications by indoor localization, e.g., indoor location-based service (ILBS), internet-of-thing (IoT), ambient assisted living (AAL), indoor emergency responder navigation, smart building (energy efficiency's occupancy detection) make the research moves forward [14]. Some technologies support indoor localization research, especially RF-based, as shown in Table 6.1.

Table 6.1 RF-based technologies for indoor localization.

Technology	Accuracy (m)	Coverage (m)	Power (watt)
GPS	1 – 20	Global	500
Radio-frequency identification (RFID)	1	1 – 50	0.02 – 0.3
Ultrawideband (UWB)	< 0.3	< 300	0.03
Wi-Fi	1 – 5	< 100	0.5 – 1
Bluetooth low energy (BLE)	1	< 10	0.01
ZigBee	1 – 5	< 30	0.02 – 0.04
GPS	1 – 20	Global	500

We focus on using a Wi-Fi-based device as it is widely available, low-cost, and simple to implement. Wi-Fi has an RSSI parameter which is easy to extract. RSSI shows the linear relationship between the distance between transmitter and receiver with its corresponding distance. However, the drawbacks of RSSI are fluctuated values in time, low precision, and the need for optimization. This drawback can be handled by fingerprint-based indoor localization [18], [22].

The fingerprint technique applies spatial locations based on location-dependent measurable features (location fingerprints and corresponding localization parameters). As we consider the RSSI, the features or parameters are stored in each fingerprint location as a fingerprint database. The fingerprint technique mainly has two phases:

1. ***Offline phase***: where we store the parameter, i.e., RSSI, from specific reference points (RPs) in the exact fingerprint location preliminary designed.
2. ***Online phase***: we locate the target based on its RSSI parameter by comparing it to those in the fingerprint database.

The whole process of the fingerprint technique is illustrated in Figure 6.1. The pattern-matching algorithm can be as simple as Euclidean distance or machine/deep learning (ML/DL) based. Considering many benefits of DL, we consider employing discriminative model of DL for the pattern-matching algorithm.

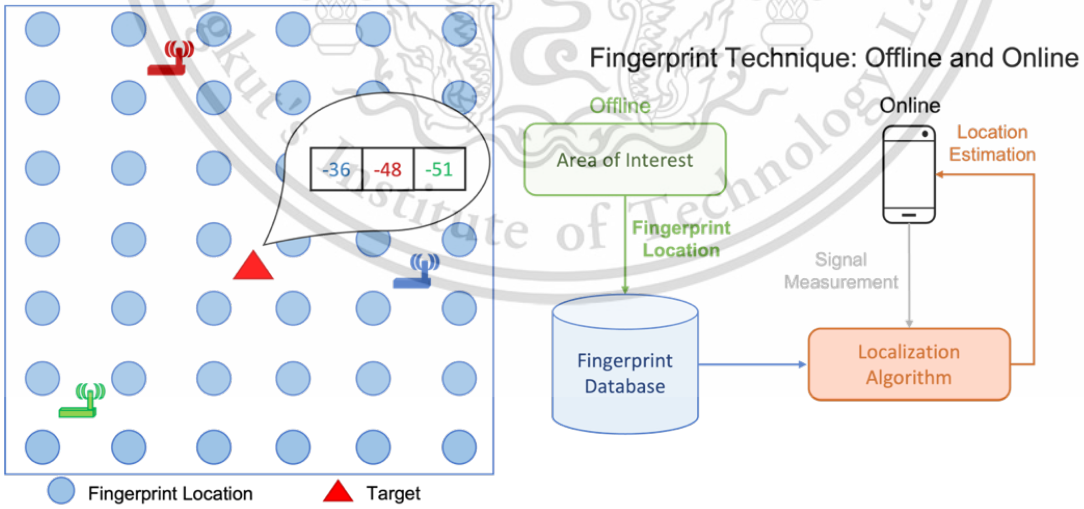


Figure 6.1 Fingerprint technique's offline and online phase.

6.3.2 Deep Learning (DL)

Figure 6.2 shows the DL's taxonomy. DL is a subset of machine learning (ML) and a subset of artificial intelligence (AI) in the broader context. DL is designed especially for feature learning, allowing meaningful information extraction from raw data without predefined features. The baseline of discriminative models is the multilayer perceptron (MLP) which we considered for our proposal on the classification task. At the same time, the latent variable-based generative model of GANs is used for RSSI synthesis.

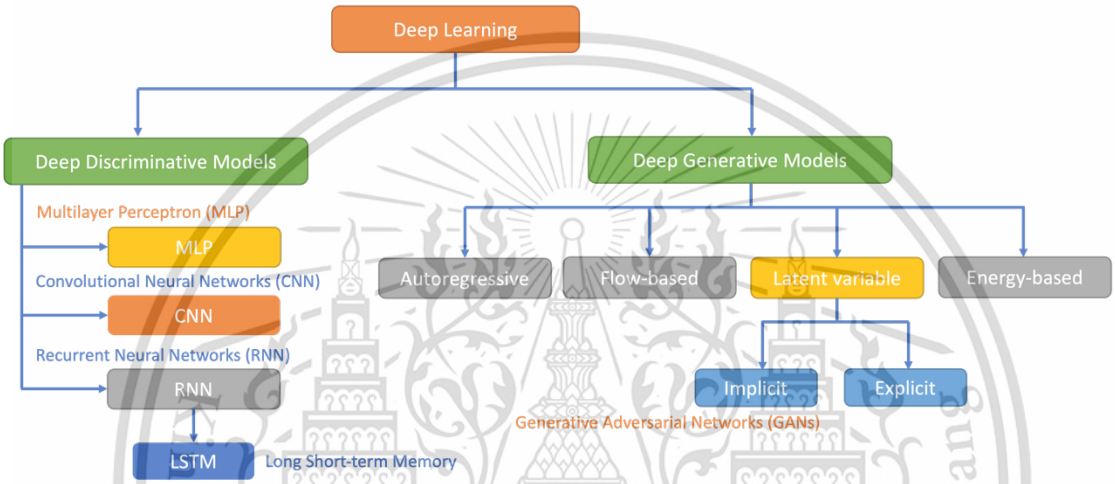


Figure 6.2 Taxonomy of DL [63], [66].

6.3.2.1 Deep Discriminative Model: MLP

MLP is an example of a deep discriminative model. It is a feed-forward deep neural network; as its name suggests, the algorithm's workflow is from input to output via hidden layers, as shown in Figure 6.3.

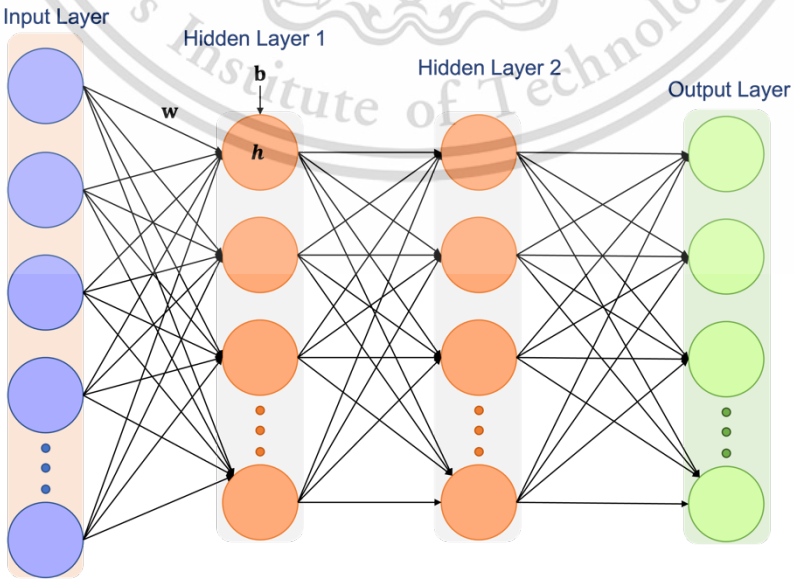


Figure 6.3 MLP architecture illustration.

MLP is primarily used in pattern recognition, prediction, and classification. The flow of data processing: the input layer receives the input signal, the hidden layers which are computational taking place are located between the input and output layer, and the output layer performs prediction and classification. There is backpropagation learning in the MLP neurons, approximating the continuous function, and solving nonlinear separable problems by having an activation function in each layer, e.g., ReLU, sigmoid, SoftMax, tanh. This activation function defines the sum of weighting and decide whether a neuron will be activated or not. The sole purpose is to add the non-linearity into output of neuron. In our proposed method, the MLP use for classification of the RSSI in same class and for constructing the generator and discriminator networks in GANs [63], [178].

6.3.2.2 Deep Generative Model: GANs

GANs is first introduce by Ian Goodfellow, et al., and involve the dueling of two networks: the Generative, G, and the discriminative; D [172]. G wants to minimize the error between fake and actual data, while D wants to maximize the capability to distinguish between the fake and the actual data. As the name suggested, GANs apply the adversarial between these two networks. GANs are illustrated in Figure 6.4.

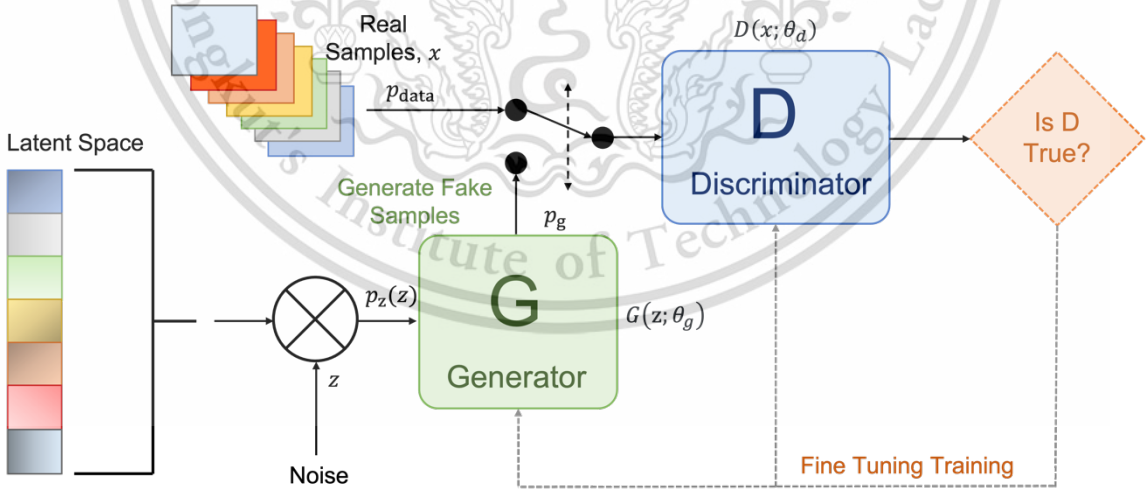


Figure 6.4 GANs architecture illustration (redrawn from [179]).

From Figure 6.4, let x be the actual data point, and its distribution is expressed by p_{data}, as the latent variable-based model; initially, the G has input noise, z, with its distribution given by $p_{\mathrm{z}}(z)$. The G learns from differential function, $G(\mathrm{z};\theta_g)$ with a

learned parameter of θ_g using MLP. $D(x;\theta_d)$ gives the function of D, which gives the probability that the output of D comes from x rather than from G distribution, p_g. D is trained to maximize $\log D(x)$, while G minimizes $\log\big(1-D(G(z))\big)$. GANs is a two-minimax game, illustrating G as the thief who wants to produce the counterfeit banknote, while D is a detective. The GANs' objective function can be expressed by Equation (6.1):

$$\min\max V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}\big[\log\big(1-D(G(z))\big)\big]. \qquad (6.1)$$

- Equation (6.1) consists of two parts, with the goal of the Discriminator being to maximize the probability of correctly assigning labels to the actual and fake data.
- Both G and D are differentiable functions. G learns to map the latent noise $\mathbf{z}\sim p_z(z)$ to the actual data distribution, $\mathbf{x}\sim p_{\text{data}}(x)$, denoted via the $G\big(\mathbf{z},\theta_g\big)$ structure, where θ_g indicates the parameters of the MLP in the Generator.
- D learns to distinguish between actual and fake data denoted via the $D(\mathbf{x},\theta_d)$ structure, where θ_d indicates the parameters of MLP in the Discriminator. The Discriminator has a binary classification structure in its output, in which 0 and 1 indicate fake and actual data, respectively.

6.4 Data Collection and Algorithm Implementation

6.4.1 Measurement Campaign

6.4.1.1 Measurement System and Setup

Figure 6.5 and 6.6 shows the system's design and actual measurement campaign, respectively. We have 8 (eight) reference points (RPs) and 25 fingerprint locations for the measurement campaign and employed a Wi-Fi-based device, ESP32. Our measurement system follows the WSNs' star topology, with the PC sending a command via the server node. The server node then communicated with the sink node. The sink node broadcasts the RSSI request to 8 RPs, and they send the RSSI to the sink node. The sink node finally sends back RSSI values to the server and finally collects them to the PC. We utilize a computer with an internal graphical processing unit (GPU) and the Jupyter Notebook on Python 3.9 for algorithm implementation.

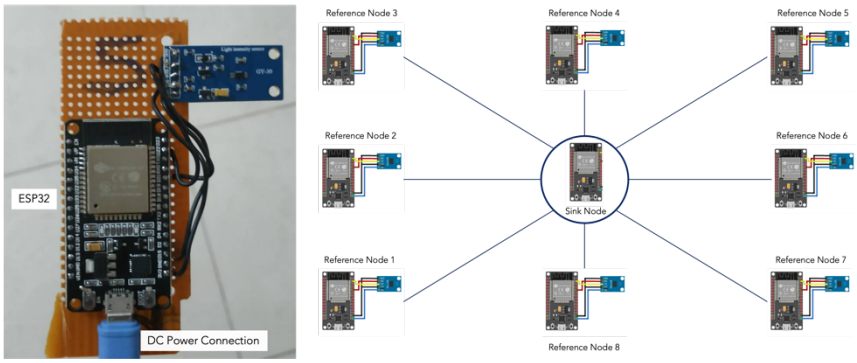


Figure 6.5 System's design.

Figure 6.6 Actual measurement campaign.

6.4.1.1 Data Structure

From measurement campaign, we obtained our RSSI values database. The total RSSI data for 25 fingerprint locations is 1,250 rows x 8 columns of RSSI values (from Node 1 to Node 8). The last two columns are the label of coordinates in 2D (x and y). The measurement's data structure is depicted in Table 6.2.

Table 6.2 RSSI data structure.

No.	RSSI_1 (Nd. 1)	RSSI_2 (Nd. 2)	RSSI_3 (Nd. 3)	RSSI_4 (Nd. 4)	RSSI_5 (Nd. 5)	RSSI_6 (Nd. 6)	RSSI_7 (Nd. 7)	RSSI_8 (Nd. 8)	Coord. (x, y)
0	-52	-56	-58	-67	-65	-64	-65	-60	(1, 1)
1	-52	-55	-56	-64	-63	-67	-68	-55	(1, 1)
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
1250	-67	-65	-69	-58	-43	-57	-69	-73	(5, 5)

*Nd = Node.

6.4.2 Algorithm Implementation

Figure 6.7 shows the workflow of algorithm implementation. There are two significant steps: the first step is to classify the data test with only the actual dataset for training. While the second step, before entering the classification by MLP, GANs are used to generate synthetic data, which is then combined to the actual dataset. Both classification results will be compared to observe how augmented data can improve localization performance.

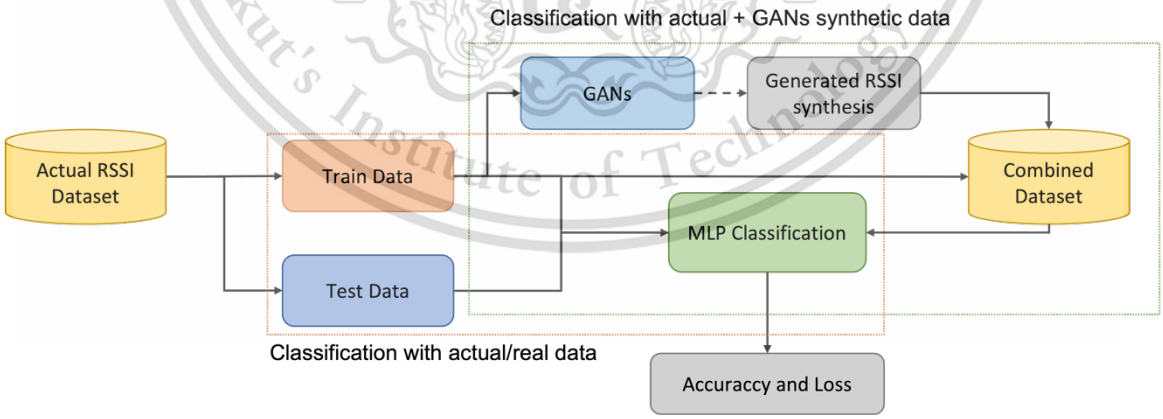


Figure 6.7 Proposed method implementation workflow.

6.4.2.1 MLP Architecture

MLP is used as a classifier for every 25 locations; its architecture is depicted in Figure 6.8. We designed the MLP as follows; the input is dense layers with 8 (RSSI based on

the number of RPs); all activation functions are ReLu except the output activation function, which is a multi-class label, SoftMax (25 labels).

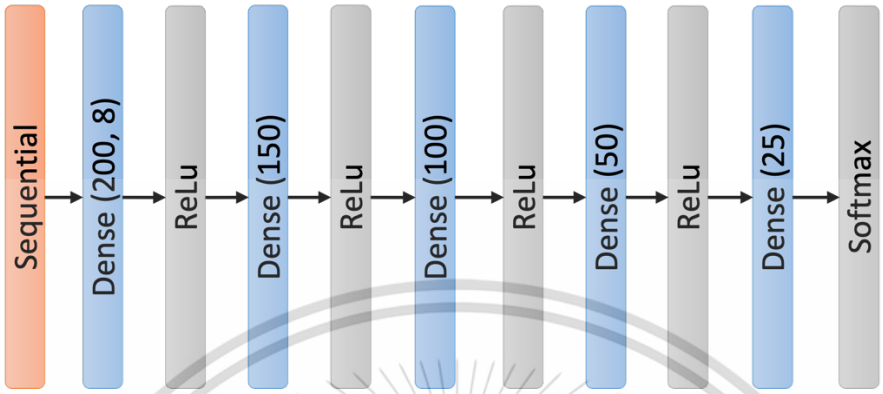


Figure 6.8 The MLP architecture.

6.4.2.2 GANs Architecture

Figure 6.9 shows the GANs architecture for both Generator and Discriminator. The layer is the same using Dense layers, LeakyReLu instead of ReLu, with 0.2 negative steps (default). As the output activation function, Generator is multi-class, so SoftMax is used, while Discriminator has binary output, Sigmoid is used. We set each layer's size according to the input size.

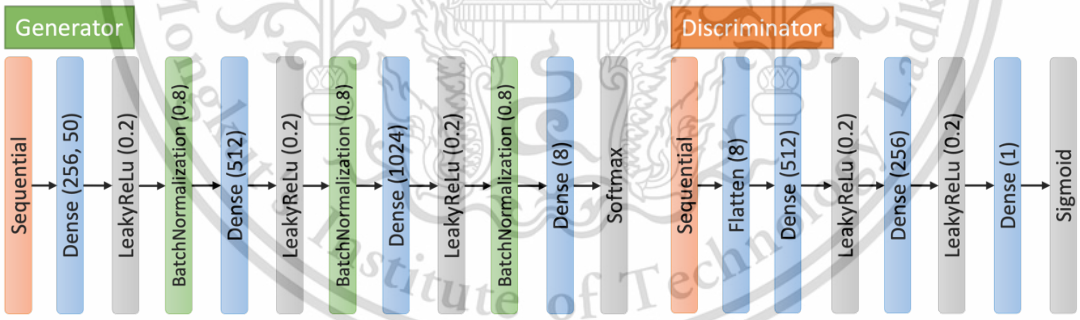


Figure 6.9 The GANs architecture.

6.4.2.3 System Model

The system model of this approach is based on [76]. We make a difference in terms of class dataset point-of-view due to the small dataset. We refer to “class” as the fingerprint locations, not a classroom. Let $F(\mathbf{x}, \theta_C)$ with $\mathbf{x} \epsilon \mathbb{R}^{1\times M}$ is a column vector of RSSI values, with M is 8 RNs. θ_C is parameter of deep learning during training and

C represent the class output nodes. In vector form, the system model equation can be written as

$$\mathcal{L}(\theta_C) = -\sum_{i=1}^{N} \mathbf{y}_i \log \hat{\mathbf{y}}_i^T \tag{6.2}$$

$\mathbf{y}_i \epsilon \mathbb{B}^{1\times C}$, $\mathbb{B} \epsilon \{0, 1\}$. $\mathbf{y}_i$ is one encoded, if 1, we will have $[1 \quad 0 \quad 0 \quad \dots \quad 0]_1, [0 \quad 1 \quad 0 \quad \dots \quad 0]_2, \dots \dots [0 \quad 0 \quad 0 \quad \dots \quad 1]_C$ and $\hat{\mathbf{y}}_i$ is a real number between 0 and 1 predicted by the deep model. The log function satisfies the hard selection of a single class.

Backpropagation is used to train the system via minimizing the log-likelihood cost function. We utilized the Adam optimizer to obtain $F(\mathbf{x}, \theta_C)$, such that the maximum value of the outputs of C nodes constitute the class predicted for a given input vector by the trained system model.

The idea is to observe how the synthetic RSSI added to fraction variation of the actual dataset can enhance the performance compared to only using the actual data. The RSSI dataset, $\boldsymbol{R}_c$, can be described as

$$\boldsymbol{R}_c = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1M} \\ r_{21} & r_{22} & \cdots & r_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ r_{K1} & r_{K2} & \cdots & r_{KM} \end{bmatrix}. \tag{6.3}$$

With r_{ij} is the magnitude of the ith observation from the jth reference node. Each column of the matrix constitutes a distribution over the desired class. We define $\mathbf{x} \epsilon \mathbb{R}^{1\times M}$ such that the generator's goal is to map prior noise latent variable $\mathbf{z} \epsilon \mathbb{R}^{1\times L}$ to the distribution of R_c. The process for producing the synthetic data is based on the cost function in Eq. (1). Convergence occurs when $D(\mathbf{x}, \theta_d) = 0.5$, meaning that the discriminator can no longer distinguish between actual and synthetic data. After this, the generator is ready to produce synthetic samples for the desired class via the same prior noise distribution $\mathbf{z}\sim p_{\mathbf{z}}(z)$.

We consider classification via MLP used for training by combining actual and synthetic RSS data. Thus, the full set of RSSI ($\mathbf{FR}$) data, which consists of both actual RSSI ($\mathbf{A}$) and synthetic RSS ($\mathbf{SR}$) data for the desired class C, can be defined as follows:

$$\mathbf{FR}_c = \begin{pmatrix} \mathbf{A_c} \\ \mathbf{SR}_c \end{pmatrix} \tag{6.4}$$

where $\mathbf{A_c} \in \mathbb{R}^{K \times M}$, $\mathbf{SR_c} \in \mathbb{R}^{Q \times M}$, and $\mathbf{FR_c} \in \mathbb{R}^{(K+Q) \times M}$, and Q is the number of added synthetic samples to the class c. In our case, Q is two-times number of fractions of actual dataset use.

6.4.2.4 GANs Implementation in Keras

We implemented GAN for our dataset on Python 3.7 using Keras (https://github.com/fchollet/keras). The pseudo code for this implementation is shown:

GAN implementation: *Pseudo Code*

Input: RSSI dataset

Output: RSSI actual + synthesis (full RSSI dataset)

1. Keras, TensorFlow, and other libraries
2. Data parameter (25 labels), MLP parameter, GAN Parameter
3. Set the percentage of actual and data to generate (synthesis)
4. Import dataset and data pre-processing (data frame, drop)
5. *MLP classification with actual data*
6. Generating synthetic data: Generator and Discriminator
7. GAN training
8. *MLP classification with actual + synthetic data*
9. Save results

The standard procedure begins with the importing libraries needed. Then, the parameters are defined, e.g., the number of labels, 25, as we discussed related to each fingerprint locations as a label. MLP parameter is set, including the number of iterations (epochs) and validation splitting. GAN parameter includes the number of latent dimensions and epochs for GAN training.

The percentage or fraction of data use is set in the beginning. Dataset is imported and pre-processed for the next stage. Then, MLP classification (only actual data) is

utilized to observe the system performance by comparing the training and validation for accuracy and loss. $\left({}^{N_{true}}/_{N_{total}}\right) \times 100$ calculates accuracy. N_{true} is the number of correct predictions, while loss is calculated by Eq. (2). The next step is to build GAN (D and G networks) to produce the synthetic RSSI data. We considered having 125 and 625 additional samples for each class by RSSI synthesis. After GAN training, the actual + synthetic data classification procedure is similar to the MLP classification for only actual data usage. Save results containing accuracy and losses both for training and validation, respectively.

As our dataset contains 1,250 samples collected from 8 reference nodes, for training and testing, we divide this dataset by half. We focus on training and validation both for accuracy and testing. For every fingerprint location that we label as "class," we have 25 classes, and each class then has 25 RSSI samples. We evaluate the actual percentage data use compared to actual and synthetic data performance. We observed that the increment data of 10% compensated it with the synthetic data we set to 125 and 625 samples. We expected that by having more synthetic data, even if we only use a small percentage of training data (actual RSSI values), the accuracy and loss performance should be comparable to or even better than those that use 100% of only actual data.

Table 6.3 shows the GANs parameters setting.

Table 6.3 GANs parameter setting.

Hyperparameter	Values
Number of Data Parameters	25
Data Shape	8 x 1
MLP Parameters: Epochs and Validation Split	20 and 0.1
GAN Parameter: Latent Dimension and GAN Epochs	50 and 2,500
Generator: Dense Layer	3, Activation Function: Leaky ReLU, Alpha: 0.2
Discriminator: Dense Layer	Same with Generator, Different Size of Dense Layer
Learning Rate	0.0002

Normalisation Technique	Batch Normalization with Momentum: 0.8.
Training Batch Size	128
Loss Function	GAN Loss
Optimizer in Generate Data Process	Adaptive Moment Estimation (Adam)[180]

6.5 Results and Discussion

We discuss our findings on the performance comparison of increment of actual data usage compared to the full set of RSSI data. As in Equation (6.4), the full RSSI set contains the actual and synthetic data. Then, the training and validation for accuracy and loss for the actual and full set of RSSI. The synthetic database exploration, including the description of actual vs. synthetic data distribution, will also be discussed. Finally, we discuss it related the computational time.

6.5.1 Actual and Full Set of RSSI Performance Comparison

The basic idea of GANs implementation for RSSI synthesis is how we can employ small datasets and still have acceptable accuracy. In this result, we compare the percentage actual and synthesis data combination. We considered adding the actual data in a 10 % increment, while the sample generation of synthetic data is 125 and 625 samples. These numbers are selected based on the number of synthetic data generated in the final 3 and 11 times greater than the actual data size. We expect that increasing the number of data will increase the performance of the MLP in classifying data.

Figure 6.10 shows that the accuracy is low when only actual data applied to MLP are used. Although all data training considered was used, the mean accuracy is 0.53 (53 %). Thus, by adding the synthetic data, as observed, e.g., by only using 20 % actual data and combining with the synthetic data both 125 and 625 samples from GANs, the mean accuracy is 0.54 (54 %), slightly higher with less use of actual training data.

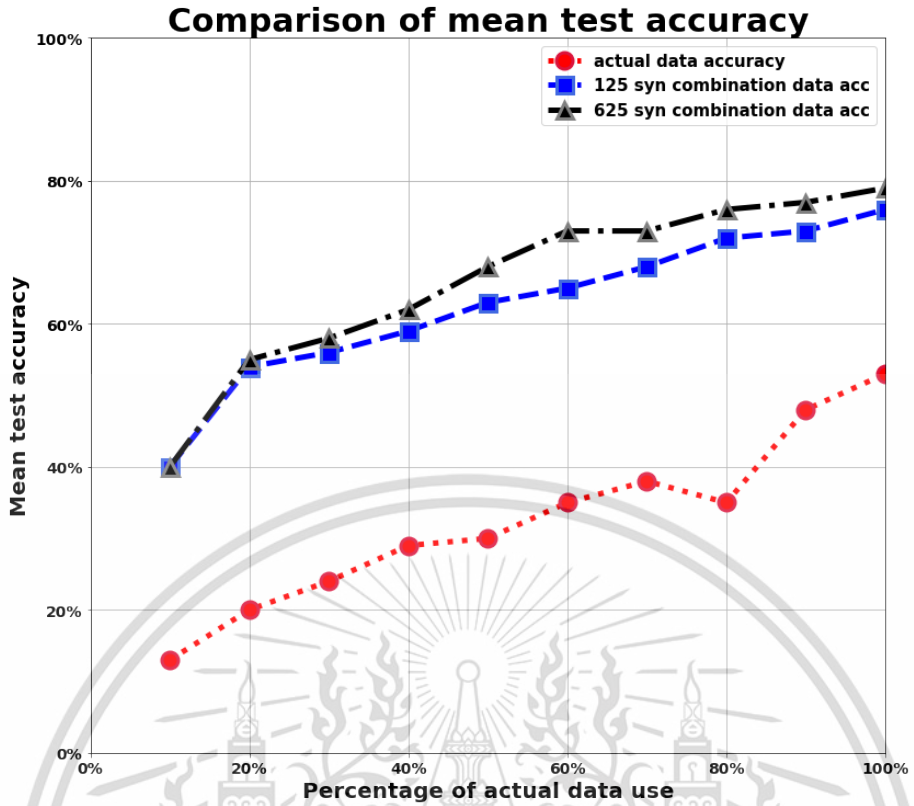


Figure 6.10 Mean accuracy: actual vs. full set of RSSI.

If using only 60 % of the actual data, the mean accuracy and loss are 0.35 (35 %) and 2.41, respectively. Then, we added the synthetic data of 125 RSSI samples from the 60 % training data by GANs and combined it to the data as the data combination of 15 actual and 125 synthetics for each class. From this new combination dataset, we got the mean accuracy and loss of 0.65 (65 %) and 1.36, respectively. The improvement in accuracy is 1.85 times higher and 1.7 times better than the loss from using actual data only. Furthermore, when we added the synthetic to 625 samples to each class, the mean accuracy improved to 0.73 (73 %) or twice better accuracy than using only 60% of the actual data for training. The loss performance also improved almost two times (1.95x) better.

Interestingly, when we used 100 % actual data only, we obtained the accuracy and loss of only 0.53 (53 %) and 1.76, respectively. Compared to the 60 % actual data and combined with 125 samples of synthetic data, we obtained a much improvement in accuracy and loss for 0.65 and 1.36, respectively. So, using only 60 % data and a synthetic combination, we can achieve a better performance by about 1.2 times. Thus, the synthetic data method by implementing GANs can be used to tackle data scarcity

and alleviate the burden in offline database construction by taking only the small dataset for the fingerprint database.

Moreover, as expected, when we added the 100 % usage of actual data with the synthetic data, e.g., adding the 125 samples of synthetic data, the mean accuracy improved to 0.76 (76 %) and loss of 0.85. When increasing the number of synthetic data samples to 625, the mean accuracy becomes 0.79 (79 %), and the mean loss is 1.03.

Table 6.4 shows that the mean accuracy performance is improved by adding synthetic data. However, when adding the 645 samples, some loss performance decreased. This high number of additional samples means that the synthetic data have some optimal numbers. Here, we are not yet considering the optimal number. However, seeing some improvement by adding 125 samples, it is accepted that that number of samples can be applied. An additional 125 samples to each fingerprint location/class give better performance in accuracy and loss in all the increments of data percentage used. Thus, the synthetic dataset became 3,125 samples (almost three times larger than the actual data; 1,250).

Table 6.4 Performance Comparison.

Real	20%		60%		100%	
Synthetic	Acc.	Loss	Acc.	Loss	Acc.	Loss
0	20 %	3.04	35 %	2.41	53 %	1.76
125	54 %	2.52	65 %	1.36	76 %	0.85
625	55 %	4.26	73 %	1.23	79 %	1.03

6.5.2 Training and Validation: Accuracy and Loss

Our dataset is 10 % smaller than our primary reference dataset [76]. However, the trend is similar; when we combined this small dataset with the synthetic data, we generated by GANs, the performance of both accuracy and loss in classification is comparable to even better than using all the actual training datasets.

As an additional number of samples of 625 is giving marginal improvement, we will consider only the synthetic data sample of 125 in the following discussion. Figure 6.11 shows how synthetic databases can improve performance in training and validation, accuracy, and loss. The 20 % actual data usage performed will be compared with the

synthetic data combination. Naturally, when we added more data for training, the accuracy and loss performance improved. The benefit of additional synthetic data is that we can start improving only by 20 % and later for actual data usage. Let us compare the 60 % and 100 % data usage as shown in Figure 6.12 and 6.13.

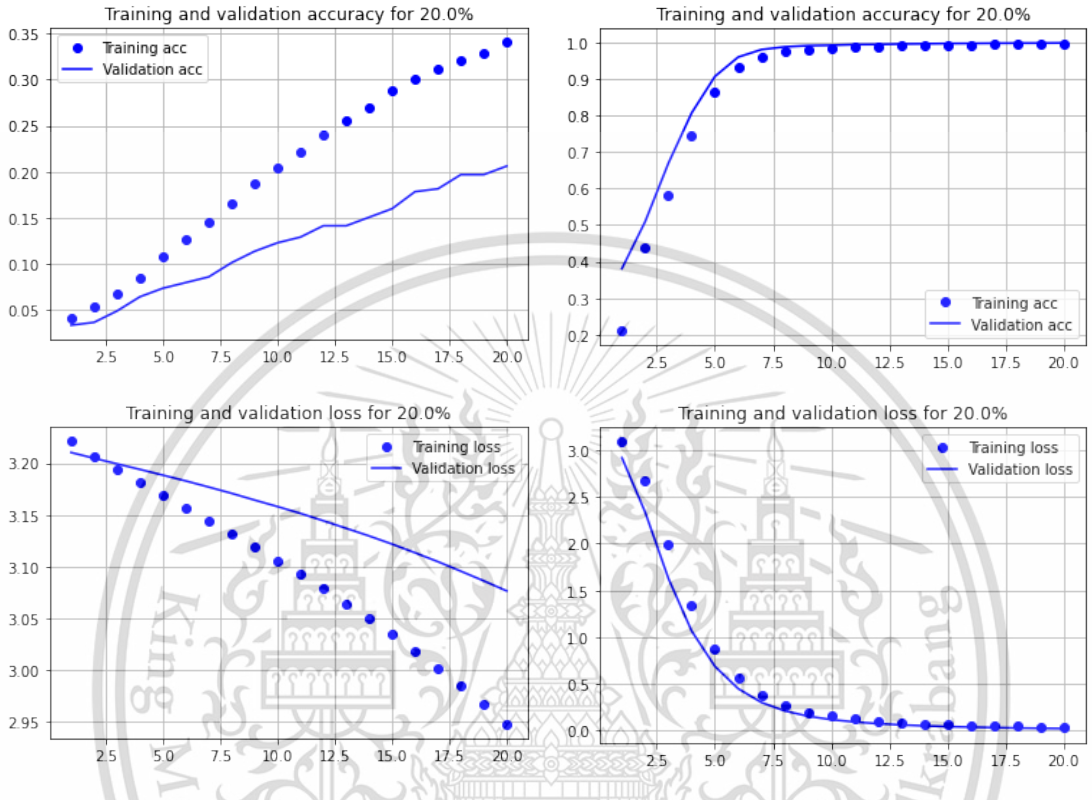
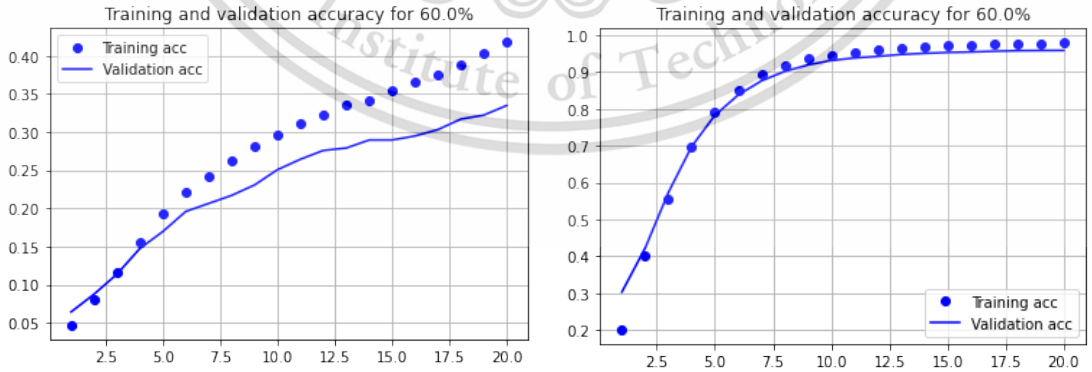
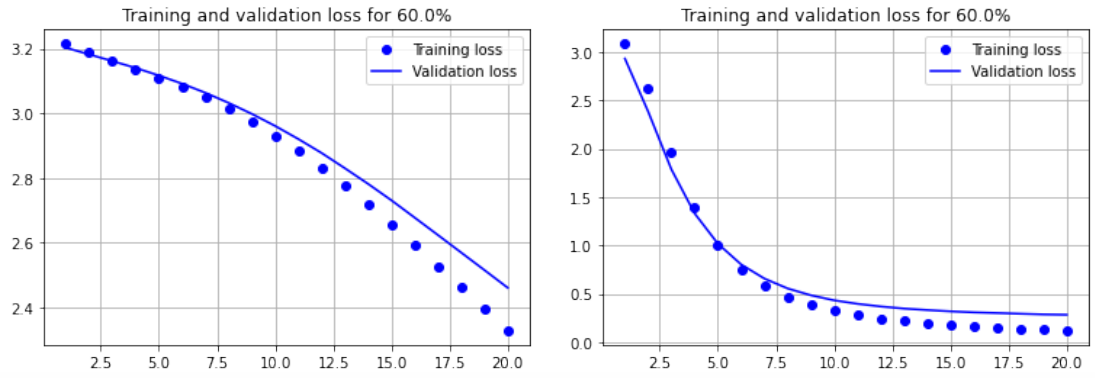


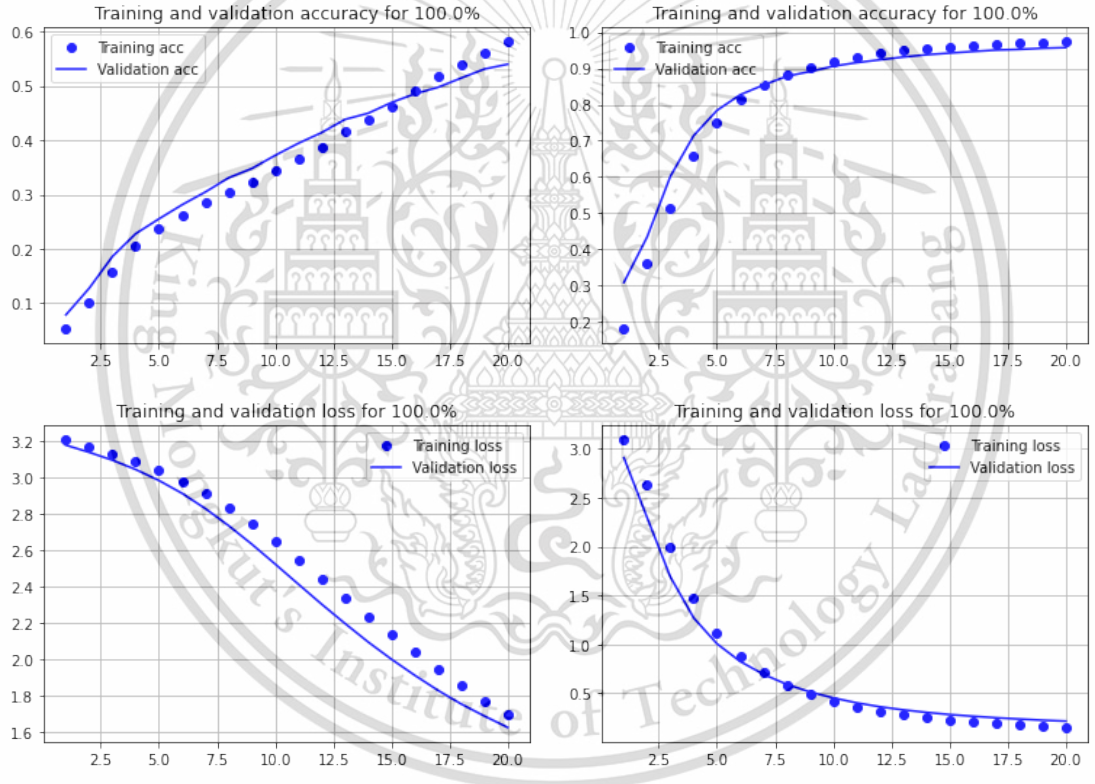
Figure 6.11 Training and validation: accuracy and loss for (a) actual vs. (b) full set of RSSI (20 % actual data).





(a) (6)

Figure 6.12 Training and validation: accuracy and loss for (a) actual vs. (b) full set of RSSI (60 % actual data).



(a) (6)

Figure 6.13 Training and validation: accuracy and loss for (a) actual vs. (b) full set of RSSI (100 % actual data).

6.5.3 Synthetic RSSI Exploration

Our dataset is 10 % smaller than our primary reference dataset [76]. However, the trend is similar; when we combined this small dataset with the synthetic data, we

generated by GANs, the performance of both accuracy and loss in classification is comparable to even better than using all the actual training datasets.

We need to explore the description of the actual and synthetic database, e.g., the mean, standard deviation (std), max, and min values. In this example, we consider exploring within the 20 % data usage. We observed how its data distribution properties differ or are similar between the actual and the synthesis. This discussion evaluated the RSSI values from AP1 or reference node one as an example.

From Figure 6.14 and 6.15 in the AP1 description, we can observe that the mean, std, max, and min values both for actual and synthesis are relatively similar. For instance, the mean RSSI and std discrepancies are -0.13 dBm and -0.27, respectively. As the discrepancy is relatively low, the synthetic data distribution is similar to the actual data. We concluded that GANs implementation is working as expected for our RSSI synthesis.

	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8
count	1251.000000	1251.000000	1251.000000	1251.000000	1251.000000	1251.000000	1251.000000	1251.000000
mean	-64.960032	-63.167066	-60.547562	-58.127098	-59.784972	-57.535572	-60.647482	-60.400480
std	5.318947	6.041727	5.283023	5.824005	5.651276	6.610366	4.870280	6.999221
min	-81.000000	-83.000000	-79.000000	-75.000000	-79.000000	-78.000000	-79.000000	-80.000000
25%	-68.000000	-67.000000	-65.000000	-61.000000	-64.000000	-62.000000	-64.000000	-66.000000
50%	-65.000000	-63.000000	-60.000000	-58.000000	-60.000000	-57.000000	-60.000000	-61.000000
75%	-61.000000	-59.000000	-57.000000	-55.000000	-56.000000	-54.000000	-57.000000	-55.000000
max	-51.000000	-45.000000	-43.000000	-35.000000	-43.000000	-35.000000	-48.000000	-43.000000

Figure 6.14 Statistical description of the actual database.

	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8
count	3125.000000	3125.000000	3125.000000	3125.000000	3125.000000	3125.000000	3125.000000	3125.000000
mean	-64.828244	-62.345525	-61.133501	-58.000828	-60.079692	-57.449583	-60.877659	-59.826856
std	5.048397	6.052459	5.401121	5.802204	5.935214	6.757984	4.302743	7.068879
min	-78.900010	-79.999985	-73.999985	-71.948730	-72.999000	-76.999710	-71.853165	-75.000000
25%	-67.992760	-65.883120	-64.962395	-61.107315	-64.904590	-62.175540	-63.999970	-65.377090
50%	-64.158700	-63.071170	-60.954536	-57.866352	-60.001800	-57.422390	-60.377464	-58.998490
75%	-61.133244	-58.595700	-57.458595	-54.139410	-56.091064	-53.599106	-57.712220	-53.904663
max	-52.000103	-46.000000	-45.000000	-41.698257	-44.387844	-36.000000	-51.000000	-45.380405

Figure 6.15 Statistical description of the synthetic database.

Figures 6.16 and 6.17 show the RSSI distribution for AP1 in both actual and synthetic data, respectively. The x axis shows the RSSI values in dBm and y axis as the probability

density. We can observe that the RSSI values are seen to be normally distributed. Both also range from around -80 dBm to -50 dBm, while the peak RSSI value is around -6 dBm.

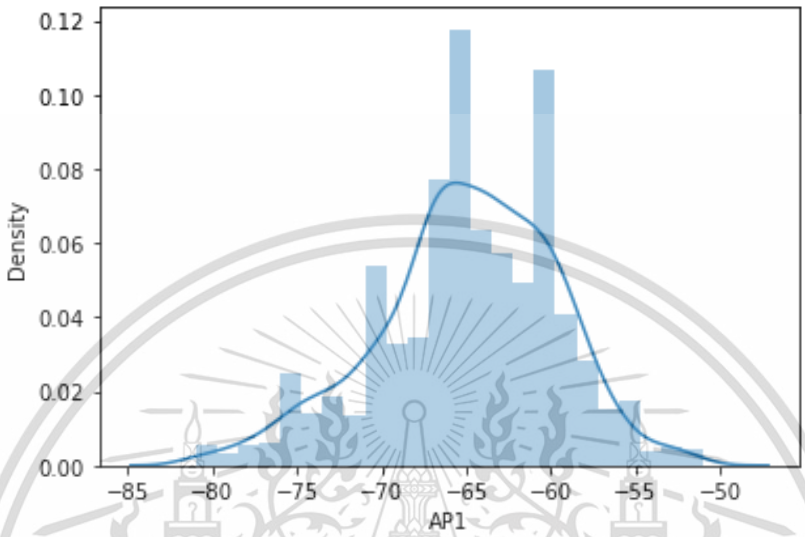


Figure 6.16 Actual RSSI distribution for AP1.

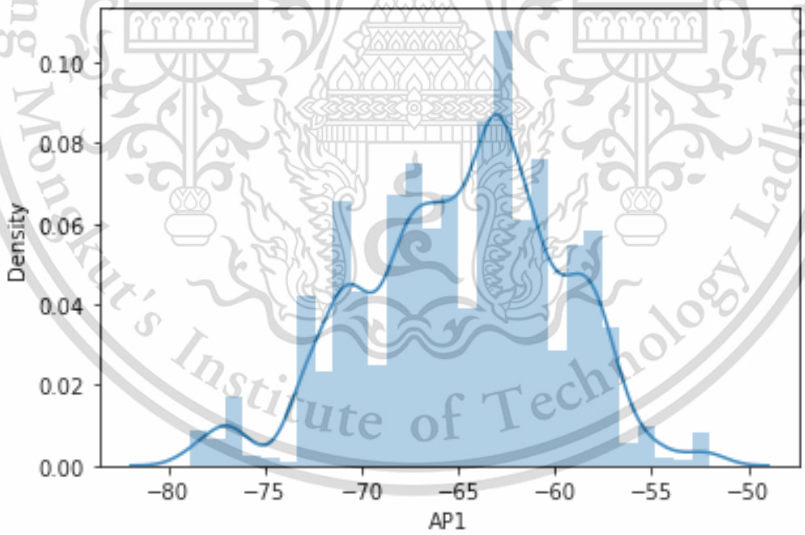


Figure 6.15 Synthetic RSSI distribution for AP1.

6.5.4 Computational Time

GAN is an iterative method to generate the samples. By setting the hyperparameters as in Table 6.5, we obtained the varied computational time considering the number of actual data and the number of samples that were produced. For each iteration for 25 classes, the iteration time ranges from 124 s to 200 s. Thus, considering the actual

percentage and number of synthetic samples took around from ~50 minutes to ~90 minutes. These results are based on the PC and its specification stated in Table 6.6 and considering both hyperparameters of Table 6.3 and Table 6.5.

Table 6.5 MLP parameter setting.

Hyperparameter	Values
Number of Input Neurons	8
Number of Hidden Layers	3
Number of Neurons in Hidden Layer	300
Number of Output Neurons	1
Activation Functions of Hidden Layers	ReLU
Activation Functions of Output Layers	SoftMax
Learning Rate	0.0002
Loss Function	Categorical Crossentropy
Optimiser	Adaptive Moment Estimation (Adam)[180]

Table 6.6 Measurement Devices and Tools.

Name	Device	Specification	Use
Reference, server, sink (target) node	ESP32 Toolkit	IEEE 802.15.11 standard, Memory 520 kB	Transceiver for RSSI values
Software	Arduino IDE, Jupyter Notebook	1.8.5 version (for windows), Python 3.7, TensorFlow, Keras	ESP32 basic program, algorithm implementation
Personal Computer (PC) 1	CPU (AMD Ryzen 5 3600) GPU (NVIDIA GeForce GTX 1660)	RAM 16 GB	For collecting the data, program ESP32

Personal Computer (PC) 2	CPU (AMD Ryzen 9 3900X), Windows 10 (64-bits), GPU (NVIDIA GeForce RTX 2080)	12-Core Processor, 3.79 GHz. RAM 32 GB, VRAM 8 GB and shared memory 16 GB (total 24 GB)	Algorithm implementation

6.6 Conclusion

We presented the effort to alleviate the drawbacks in offline database fingerprint construction by generating a synthetic database. This paper focused on how the deep learning approach can improve a small fingerprint database containing RSSI values. GAN has successfully generated the synthetic RSSI data. Both actual and full sets of RSSI have been compared by applying MLP to test the class output; in our case, the fingerprint location with a total of 25 locations. Our dataset is relatively small, by seeing the results of MLP classification, even by using all data, the accuracy is still relatively low (53 %).

On the other hand, by combining 20 % of actual data with its synthetic RSSI values, the accuracy is 0.54 (54 %). Moreover, when we used only 60 % of the actual data and combined them with the 625 generated RSSI samples for each fingerprint location, we can improve both performance accuracy and loss, with an accuracy of 0.73 %. As discussed earlier, the additional percentage of actual data improved the performance, naturally. We can apply synthetic data generated by GANs to consider less data when implementing fingerprint-based indoor localisation for real applications.

Chapter 7
General Discussion and Conclusions

The basic idea of our “database enhancement” is to enhance both in quality (uniqueness) and quantity by database synthesis for the radio fingerprint database. We have proposed the classical and deep learning-based method to answer the challenge of the database construction issues in the fingerprint database. The enhancement in number both for localization purposes and training purposes has been proven beneficial. The enhancement in localization performance results has also been visibly impactful. Chapter 7 of this dissertation is to wrap up what we have presented in the previous 6 (six) chapters. This chapter discusses the overall finding in our proposed radio fingerprint database enhancement via classical and deep generative models, continued by the conclusions that summarize the whole dissertation. Finally, we discuss the prospects and development in the area of fingerprint-technique-based indoor localization database enhancement.

7.1 General Discussion of Radio Fingerprint Database Enhancement

The main issue of the radio fingerprint-based indoor localization is the burden of constructing its database in the offline phase. We have presented our approach to alleviating the database issue by radio fingerprint database enhancement using several methods: dual-features/parameters, classical interpolation, and deep generative models-based for RSSI synthesis. We want to discuss further our method results with corresponding future works that might be visible for this particular research topic.

Dual-Features for IDFL

We found limited literature on dual or multi features fingerprint databases for IDFL, including the ML algorithm for estimating position. We found some difficulties validating the fingerprint database for empty and occupied rooms (the condition in which a room has a person standing at each fingerprint location). As we observed, the values are similar in both datasets. This condition is challenging in IDFL, especially by using the RSSI

parameter. In addition, the illuminance sensor relied only on the available lamp in that room; it is not that ideal to kind of directing the "uniqueness" in each position because, as we are aware, some fingerprint locations have similar illuminance values associated with them even if they are spread far to each other in the area of interest. However, relying on the available infrastructure is a good offering point because of its usefulness later in the implementation stage. The chosen Random Forest as the localization algorithm is justified within our data points and the nature of classification in terms of target estimation. The results also show some inconsistent results in what we initially expected, but some key improvements have been achieved in these dual features, especially for IDFL.

Classical Interpolation and Regression

The offered solution of database enhancement via classical interpolation and regression emphasizes how simple yet effective a method can be. As the fingerprint database construction has been the main pain point, how to make the database denser than the original one is interesting to discuss. In our approach of classical interpolation and regression, we found that by seeing the measurement campaign layout and fingerprint locations, we can approach simple linear or/and polynomial interpolation/regression (after observing how the RSSI fluctuated vs. distance). The proposal also worked on both 2D and 3D environments, which broadened the understanding that the system can be implemented with minimal additional cost, both computational and fee for database construction.

The RSSI values from ZigBee technology have also proven to be stable in handling the environment, as we consider in the 2D environment, the line-of-sight condition. While in a 3D environment, we also "simulated" the complex interactive objects, i.e., humans and furniture. These objects we expected these to have some effects on the RSSI values. The results also show that classical interpolation and regression on 2D and 3D cases can enhance database and localization performance.

RSSI Synthesis via Variational Autoencoders (VAEs)

In this proposal, we generated the RSSI synthesis via VAEs. The main reason is to see how the latent distribution of VAEs can synthesize RSSI. We convert our RSSI database numerical structure to images in a specific pixel size to execute this idea. Considering the 8 (eight) RSSI values from 8 reference nodes, we put those 8 RSSI values into an empty 3-

by-3-pixel grayscale image, one left pixel filled with high power (100 dBm). The grayscale degree of each pixel represented the normalized RSSI values which needed to scale from dBm values to pixel values by normalization.

Using the original VAEs, we can generate new RSSI values by selecting specific distributions in the latent distribution. As known, the VAEs have low computational complexity but pay the cost of the reliability of the synthetic results. VAEs can generate output slightly different from the input through its latent distribution. The challenge is in this point, where, in some sense, the RSSI values are similar, especially when there is a symmetry point in how we collect data. The right diagonal values could have similar values compared to the left ones (the difference is only in how we stacked them pixel-wise). Some of the latent distribution with low epoch has been piling up. We will have similar RSSI values in some fingerprint locations by choosing those distributions, making the later process difficult. The marginal results of this RSSI synthesis via VAEs allowed us to take another deep generative model: generative adversarial networks (GANs).

RSSI Synthesis via Generative Adversarial Networks (GANs)

Due to the marginal results of VAEs, we turned our attention to GANs for RSSI synthesis. Instead of converting dBm to pixels, we use the RSSI as vector noise corresponding to its values (direct use). As RSSI from VAEs could not continue to the localization stage, we consider using another deep learning (discriminative) model, i.e., MLP, for later in the localization process. GANs implementation aims to state each fingerprint location as a class or label. This consideration is based on the limited size (small fingerprint database). Other considerations involve the size of synthetic data that GANs will produce and some related to hyperparameters affecting the computational cost of GANs. In VAEs, it is known for less computational cost but needs to estimate intractable posterior. In contrast, in GANs, we do not need to estimate some distribution but to adverse two deep neural networks iteratively to update the training for producing the synthetic data.

For target classification, MLP takes both actual data and combines data (synthetic RSSI + actual RSSI), and the overall results are what we expected. First, by having more actual data, the accuracy will linearly improve. Second, the accuracy can also be improved by using only a tiny portion of actual data and combining it with the synthetic data from GANs. The apparent differences are that the MLP validation and loss performance improved by having synthetic data, leading to the improvement of target classification.

There has yet to be any discussion regarding the optimal number of actual data to support this method and the adequate number of synthetic data to generate. The latter part will discuss the effect of GANs-generated data on the accuracy and its computational cost in training. These two issues are left for future work.

7.1.1 Error Representation

Our approach in radio fingerprint database enhancement gives localization improvement. However, the IDFL approach using Random Forest gives only marginal accuracy results. Here is why: the nature of RSSI is the averaged power values, unlike CSI, whose multipath components can observe the time flight (delay), phases, and power in individual occurrences. Our approach makes it difficult to have unique properties in the location compared to conventional device-based, leading to marginal accuracy.

The error location is visible in the interpolation method by observing the actual and predicted/estimated location. RSSI prediction error in the VAEs approach is relatively high because of many factors: the small pixel representation and the difficulty of learning for similar RSSI values. Thus, the latent distribution's difference is not distinct, leading to the inaccuracy of RSSI prediction. On the other hand, GANs' error representation is based on MLP classification output from the actual testing data. Compared to some references, the accuracy values are similar when using GANs: 70-80% localization accuracy with a comparable number of synthetic data.

7.1.2 Prospective Applications

Some prospective applications from our approaches:

1. Warehouse localization. The localization in an indoor warehouse is the most possible because the error is less than in the measurement's grid, and the warehouse dynamic is minimal.
2. For IDFL, it is helpful for elderly monitoring or people in need. It is also essential in the security intruder because of non-device attachment to the targets/objects.
3. Interpolation techniques can be used in RSSI-based indoor localization, ranging from airports, museums, and schools, with probably not that precise in location but more in human recognition in some local-area or grids.

4. Robot localization. As our measurement campaigns are primarily small, we consider applying our methods also in finding the location of robots.
5. For deep learning-based, the methods can be applied in a highly dynamic environment with the need to update the fingerprint fast. The sparse fingerprint database can be used in the first place and enhanced/added by synthetic data.

7.2 Conclusions

Our first approach in the radio fingerprint database enhancement involves the dual features/parameters in one set of fingerprint databases: RSSI and illuminance values. We also highlight that employing the baseline ML model, i.e., Random Forest, can help improve the accuracy and precision of the IDFL system. From the localization training and testing results, Random Forest gives better precision represented as the STD than other ML algorithms, k-NN and NN; however, k-NN is slightly better than Random Forest in terms of localization accuracy represented by RMSE. For this classical approach in dual-feature database enhancement, we utilized the illuminance values from the attached illumination sensor and the Wi-Fi-based device, ESP32. This device provided the RSSI values and communicated and stored the illuminance sensor reading to the coordinator or sink node. We found that using dual features of illuminance and RSSI together with the ML implementation; the results were acceptable, with the system's accuracy of 1.65m better than another basic ML; neural network (NN), while precision has 0.88m and better precision compared to k-nearest neighbor (k-NN) and NN.

Unlike the device-based localization system, where the parameter values for fingerprint databases are easier to differ, the effects of the RSSI fluctuations on the IDFL system somehow see a similarity between an empty and an occupant room. The emerging limitations are the measurement campaign devices; their calibration, antenna pattern, and directiveness illuminance sensor cannot be evaluated. Thus, the data that we receive vary and fluctuate most. The fingerprint database enhancement using dual-features or multi-features by using sensor fusions may give higher performance but have a threshold in the system's complexity [16], [181], [182]. However, in our current approach to dual features for IDFL, there are more rooms to improve, i.e., evaluating and changing the measurement setup procedure for RSSI (improving the data collection method), applying another signal

parameter, e.g., CSI, adding the number of datasets, and implementing deep learning for our future works. Implementing deep learning and using CSI, which is more robust and reliable than RSSI, can improve the performance of IDFL [183].

The second approach is by applying interpolation and regression techniques. The same issue was explored to tackle the drawbacks of offline fingerprint database construction. In this approach, we have two different datasets from our previous research's actual measurement campaign. The dataset is in 2D and "simulated" 3D environments using a wooden bookshelf (4 levels). Unlike the dual-features fingerprint, in this approach, we used a device for measurement based on IoT technology which is a cost-effective and straightforward implementation of ZigBee standards. RSSI values obtained from ZigBee devices are robust against the room condition. We built the WSNs-based indoor localization and validated our approach by some scenarios; the four target type placements, namely diagonal positions both left and right, the horizontal and vertical position for the 2D environment, while three scenarios tested 3D environment; clean, human body effects, and furniture effects.

From all scenarios, almost all results agreed that enhancing the database to expand the database grids by artificial data can reduce localization prediction errors. The simple bilinear interpolation stood out as the best accuracy performance from all scenarios. For the 2D environment, it can reduce up to 0.2 m error, and 0.14 m (14 cm) for the 3D environment, in which this environment the grid size is 0.22 m (22 cm). The performance improvement by replacing and adding the synthesis or artificial database can facilitate actual implementation. Thus, the sparsity issues in the fingerprint database can be tackled, and the offline database construction will be less burden.

Then, in the third and final approach, we emphasized applying deep learning for radio fingerprint database enhancement. In the third approach, we focused on the explicit density model, i.e., variational autoencoders (VAEs). This model must estimate the intractable posterior distribution by statistical similarity by a graphical model. The origin applications of this model are broadly known in image processing or related to the images, both statics and videos. In our approach, we convert the RSSI values into a 3-by3-pixel image and are eager to synthesize them by VAEs. We proposed VAE implementation to synthesize RSSI values from actual RSSI from a relatively small measurement campaign. The results show that the latent distribution in our VAE implementation can be used for RSSI synthesis. Observing the latent distribution from all epochs shows that each cluster

of coordinates is not so distinct from others, making some RSSI syntheses inaccurate. Our number of RSSI datasets and the small image size assumption (3-by-3) could affect this high correlation between clusters. The remaining challenge is to differ the piling position in latent distribution, since like RSSI values, the similar position we record the values, the similar RSSI values.

Moreover, the different location but symmetry in the sense of fingerprint location will yield a similar RSSI with only a different structure based on its corresponding reference nodes. From this perspective, we can look at our problem's possible solution through several works: to explore denser datasets via publicly available datasets to conduct measurement campaigns and to improve the RSSI-to-image conversion assumption, including its size and other considerations.

The final approach to the radio fingerprint database enhancement by employing GANs was expected to be more reliable than the previous chapter on VAEs. So far, we have presented an effort to alleviate offline database fingerprint construction drawbacks by generating a synthetic database. The origin motivation of our database enhancement research is that we can achieve acceptable localization system performance by only using limited data or a small fingerprint database. In our final approach, we utilized the data of RSSI from low-cost Wi-Fi devices and implemented GANs for generating the synthetic RSSI data. Moreover, for target classification in each position, we employed the MLP.

We generated 125 and 625 data, representing about 10% and 50% of the total data size (1,250). From the results, by using only 50% of actual data combined with the 125 synthetic data, we can improve the accuracy by more than 200% compared to only using 50% of actual data and show a 60% improvement in the loss. By comparing the result to using only 100% actual data, the combination dataset gives better accuracy and loss performance for about 15% improvement in accuracy. Combining 100% actual data and 125 synthetic data gives the best performance of 0.76 and 0.85, respectively, achieving an improvement of 144% in accuracy and 200% in loss performance. By implementing deep learning for fingerprint techniques for data augmentation and classification, we can achieve good performance and reduce the workload of fingerprint database construction. Our challenge in this research was conducting an efficient measurement campaign regarding database construction and high performance. On the other hand, employing the deep learning algorithm for database synthesis needs high

computational power, which may need to be installed at that current time for a smartphone-based indoor localization system. This remaining problem will make future works on radio fingerprint database enhancement more interesting.

7.3 Future Developments

The findings in dual features for IDFL are its challenges on the fingerprint database, the use of Random Forest to increase the precision and the number of datasets needed for ML-based IDFL. This finding gives some implications on 1) finding alternatives to collect the dataset and 2) using different illumination sources. The first implication results in consideration of utilizing several Transmitter-Receiver couples instead of star topology communication. This new arrangement will have few features but have impactful diversity on the RSSI values (empty and occupied room). For the second implication, the available source of illumination (lamps) is acceptable and prioritized but modifying the placement in the communication system to be more direct, e.g., an additional cone in the illumination sensor, will also increase the sensing sensitivity. Once we have improved results, the research on IDFL will impact many applications, mainly that the target/object does not carry any devices.

For the classical interpolation and regression, the denser database is achieved and proven to improve localization performance both for 2D and 3D environments. Since the methods are relatively low in cost and computational, they can give implications on various applications emphasizing the use of RSSI localization parameters even on a significant scale.

For the deep generative model-based, the first approach is applying VAEs. The marginal accuracy is obtained due to issues related to the conversion of RSSI on small pixels and the latent distribution, which have a high correlation for respected sample locations. It is implied that for future work, the size of image representation needs to be expanded, or the alternatives in using RSSI for VAEs must be explored. While GANs, the synthetic RSSI can help to improve the localization performance and have prospective future development on the number of synthetic data vs. the computational cost and hyperparameters.

Overall, future development must consider several approaches if the low-cost and straightforward localization parameters are emphasized, especially for RSSI. One

obvious drawback of RSSI is its fluctuating natures, especially in some low-cost and low-power devices. The future development could be the next steps in the signal point of view if the RSSI is still the first choice for consideration:

1. We can explore the path loss model evaluation and validation to minimize the distinct difference in application, including detecting outliers in RSSI data.
2. The depth calibration of the device in the indoor environment is needed.
3. The threshold between the number of RSSI snapshots in the fingerprint locations vs. time allocation, to keep the data obtaining time efficient. The number of data will also benefit the implemented algorithm, especially DL.

The future development also can explore other radio frequency technologies, including ultrawideband (UWB). With the ranging methods, UWB can have high precision and accuracy results. However, it has impacted cost issues and the system's complexity. The combination of UWB and Wi-Fi has also been proven helpful in the cost and high-performance requirements. Other considerations can be more focused on using sensor-based localization parameters, e.g., an inertial measuring unit (IMU), magnitude, pervasive, mechanical, and lighting sensors, more preferably on availability of it in an indoor environment facility. Adaptive deep learning is also an exciting topic where the hyperparameter and learning process is automatically based on the input of the updated fingerprint database.

Reference

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process Mag*, vol. 22, no. 4, pp. 54–69, 2005, doi: 10.1109/MSP.2005.1458287.

[2] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019, doi: 10.1109/COMST.2019.2911558.

[3] S. He and S. H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1. Institute of Electrical and Electronics Engineers Inc., pp. 466–490, Jan. 01, 2016. doi: 10.1109/COMST.2015.2464084.

[4] R. Talla-Chumpitaz, M. Castillo-Cara, L. Orozco-Barbosa, and R. García-Castro, "A novel deep learning approach using blurring image techniques for Bluetooth-based indoor localisation," *Information Fusion*, vol. 91, pp. 173–186, Mar. 2023, doi: 10.1016/j.inffus.2022.10.011.

[5] M. Bouet and A. L. dos Santos, "RFID Tags: Positioning Principles and LocalizationTechniques," in *IFIP Wireless Days Conference 2008*, Dubai: IEEE, Nov. 2008.

[6] A. R. J. Ruiz and F. S. Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis," *IEEE Trans Instrum Meas*, vol. 66, no. 8, pp. 2106–2117, Aug. 2017, doi: 10.1109/TIM.2017.2681398.

[7] S. Sadowski and P. Spachos, "RSSI-Based Indoor Localization with the Internet of Things," *IEEE Access*, vol. 6, pp. 30149–30161, Jun. 2018, doi: 10.1109/ACCESS.2018.2843325.

[8] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors (Switzerland)*, vol. 18, no. 11. MDPI AG, 2018. doi: 10.3390/s18113995.

[9] R. Mautz and S. Tilch, "Survey of optical indoor positioning systems," in *2011 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2011*, 2011. doi: 10.1109/IPIN.2011.6071925.

[10] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A survey on wireless indoor localization from the device perspective," *ACM Computing Surveys*, vol. 49, no. 2. Association for Computing Machinery, Jun. 01, 2016. doi: 10.1145/2933232.

[11] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, and M. Atri, "A Survey of Localization Systems in Internet of Things," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 761–785, Jun. 2019, doi: 10.1007/s11036-018-1090-3.

[12] I. A. Getting, "The Global Positioning System," *IEEE Spectr*, vol. 30, no. 12, pp. 36–38, 1993, doi: 10.1109/6.272176.

[13] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, "A Review of Indoor Localization Techniques and Wireless Technologies," *Wireless Personal Communications*, vol. 119, no. 1. Springer, pp. 289–327, Jul. 01, 2021. doi: 10.1007/s11277-021-08209-5.

[14] F. Alhomayani and M. H. Mahoor, "Deep learning methods for fingerprint-based indoor positioning: a review," *Journal of Location Based Services*, vol. 14, no. 3. Taylor and Francis Ltd., pp. 129–200, Jul. 02, 2020. doi: 10.1080/17489725.2020.1817582.

[15] N. Singh, S. Choe, and R. Punmiya, "Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview," *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 127150–127174, 2021. doi: 10.1109/ACCESS.2021.3111083.

[16] A. Yassin *et al.*, "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2017, doi: 10.1109/COMST.2016.2632427.

[17] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A Survey of Machine Learning for Indoor Positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020, doi: 10.1109/ACCESS.2020.3039271.

[18] D. J. Suroso, F. Y. M. Adiyatma, P. Cherntanomwong, and P. Sooraksa, "Fingerprint Database Enhancement by Applying Interpolation and Regression Techniques for IoT-based Indoor Localization," *Emerging Science Journal*, vol. 4, no. Special issue, pp. 167–189, 2020, doi: 10.28991/esj-2021-SP1-012.

[19] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor Intelligent Fingerprint-Based Localization: Principles, Approaches and Challenges," *IEEE*

Communications Surveys and Tutorials, vol. 22, no. 4, pp. 2634–2657, Oct. 2020, doi: 10.1109/COMST.2020.3014304.

[20] C. Basri and A. El Khadimi, "Survey on indoor localization system and recent advances of WIFI fingerprinting technique," in *International Conference on Multimedia Computing and Systems -Proceedings*, IEEE Computer Society, Apr. 2017, pp. 253–259. doi: 10.1109/ICMCS.2016.7905633.

[21] D. J. Suroso, P. Cherntanomwong, P. Sooraksa, and J. Takada, "Location fingerprint technique using Fuzzy C-Means clustering algorithm for indoor localization," in *TENCON 2011 - 2011 IEEE Region 10 Conference*, IEEE, Nov. 2011, pp. 88–92. doi: 10.1109/TENCON.2011.6129069.

[22] D. J. Suroso, P. Cherntanomwong, and P. Sooraksa, "Deep Generative Model-based RSSI Synthesis for Indoor Localization," in *19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ECTI-CON54298.2022.9795409.

[23] D. J. Suroso, P. Cherntanomwong, and P. Sooraksa, "Synthesis of a Small Fingerprint Database through a Deep Generative Model for Indoor Localisation," *Elektronika ir Elektrotechnika*, vol. 29, no. 1, pp. 69–75, Feb. 2023, doi: 10.5755/j02.eie.31905.

[24] D. J. Suroso, P. Cherntanomwong, and P. Sooraksa, "Indoor Device-free Localization Using Received Signal Strength Indicator and Illuminance Sensor for Random-forest-based Fingerprint Technique," *Sensors and Materials*, vol. 33, no. 12, pp. 4331–4345, 2021, doi: 10.18494/SAM.2021.3632.

[25] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput Surv*, vol. 46, no. 2, 2013, doi: 10.1145/2543581.2543592.

[26] H. Li, J. K. Ng, and K. Liu, "Handling fingerprint sparsity for Wi-Fi based indoor localization in complex environments," in *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, Institute of Electrical and Electronics Engineers Inc., Aug. 2019, pp. 1109–1116. doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00210.

[27] S. Djosic, I. Stojanovic, M. Jovanovic, T. Nikolic, and G. L. Djordjevic, "Fingerprinting-assisted UWB-based localization technique for complex indoor environments,"

Expert Syst Appl, vol. 167, no. November 2020, p. 114188, 2021, doi: 10.1016/j.eswa.2020.114188.

[28] L. Zhang and H. Wang, "3D-WiFi: 3D Localization with Commodity WiFi," *IEEE Sens J*, vol. 19, no. 13, pp. 5141–5152, 2019, doi: 10.1109/JSEN.2019.2900511.

[29] Z. Deng, J. Feng, S. Wang, Z. Qu, J. Zhang, and W. Si, "An accurate and easy deployment array gain-phase error calibration method for DoA estimation in Wi-Fi network," *Ad Hoc Networks*, vol. 112, no. June 2020, p. 102355, 2021, doi: 10.1016/j.adhoc.2020.102355.

[30] X. Tian, S. Zhu, S. Xiong, B. Jiang, Y. Yang, and X. Wang, "Performance Analysis of Wi-Fi Indoor Localization with Channel State Information," *IEEE Trans Mob Comput*, vol. 18, no. 8, pp. 1870–1884, 2019, doi: 10.1109/TMC.2018.2868680.

[31] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings - IEEE INFOCOM*, IEEE, 2000, pp. 775–784. doi: 10.1109/infcom.2000.832252.

[32] Z. Xiang, H. Zhang, J. Huang, S. Song, and K. C. Almeroth, "A hidden environment model for constructing indoor radio maps," in *Proceedings - 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005*, 2005, pp. 395–400. doi: 10.1109/WOWMOM.2005.5.

[33] J. Zuo, S. Liu, H. Xia, and Y. Qiao, "Multi-phase fingerprint map based on interpolation for indoor localization using iBeacons," *IEEE Sens J*, vol. 18, no. 8, pp. 3351–3359, Apr. 2018, doi: 10.1109/JSEN.2018.2789431.

[34] H. Zhao, B. Huang, and B. Jia, "Applying kriging interpolation for WiFi fingerprinting based indoor positioning systems," in *IEEE Wireless Communications and Networking Conference, WCNC*, Institute of Electrical and Electronics Engineers Inc., Sep. 2016. doi: 10.1109/WCNC.2016.7565018.

[35] D. Mao, W. Shao, Z. Qian, H. Xue, X. Lu, and H. Wu, "Constructing accurate Radio Environment Maps with Kriging Interpolation in Cognitive Radio Networks," in *2018 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, CSQRWC 2018*, Institute of Electrical and Electronics Engineers Inc., Sep. 2018. doi: 10.1109/CSQRWC.2018.8455448.

[36] F. Y. Martin Adiyatma, D. Joko Suroso, and P. Cherntanomwong, "Fingerprint Database Enhancement using Spatial Interpolation for IoT-based Indoor Localization," in *ICSEC 2022 - International Computer Science and Engineering*

Conference 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 192–197. doi: 10.1109/ICSEC56337.2022.10049367.

[37] X. Wang, Z. Chen, S. Zhang, and J. Zhu, "Super-Resolution Based Fingerprint Augment for Indoor WiFi Localization," *2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings*, vol. 2020-Janua, 2020, doi: 10.1109/GLOBECOM42002.2020.9348146.

[38] T. Kim Geok *et al.*, "Review of Indoor Positioning: Radio Wave Technology," *Applied Sciences*, vol. 11, no. 1, p. 279, 2020, doi: 10.3390/app11010279.

[39] Y. K. Cheng, H. J. Chou, and R. Y. Chang, "Machine-learning indoor localization with access point selection and signal strength reconstruction," *IEEE Vehicular Technology Conference*, vol. 2016-July, 2016, doi: 10.1109/VTCSpring.2016.7504333.

[40] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A Survey of Machine Learning for Indoor Positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020, doi: 10.1109/ACCESS.2020.3039271.

[41] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, ACM, Sep. 2007, pp. 222–229. doi: https://doi.org/10.1145/1287853.1287880.

[42] J. Xiao, S. L. Joseph, X. Zhang, B. Li, X. Li, and J. Zhang, "An Assistive Navigation Framework for the Visually Impaired," *IEEE Trans Hum Mach Syst*, vol. 45, no. 5, pp. 635–640, Oct. 2015, doi: 10.1109/THMS.2014.2382570.

[43] T. Helten, M. Muller, H. P. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 2013, pp. 1105–1112. doi: 10.1109/ICCV.2013.141.

[44] S. Tao, M. Kudo, B. N. Pei, H. Nonaka, and J. Toyama, "Multiperson Locating and Their Soft Tracking in a Binary Infrared Sensor Network," *IEEE Trans Hum Mach Syst*, vol. 45, no. 5, pp. 550–561, Oct. 2015, doi: 10.1109/THMS.2014.2365466.

[45] M. Scholz, L. Kohout, M. Horne, M. Budde, M. Beigl, and M. Youssef, "Device-free radio-based low overhead identification of subject classes," in *WPA 2015 - Proceedings of the 2nd Workshop on Physical Analytics*, Association for Computing Machinery, Inc, May 2015, pp. 1–6. doi: 10.1145/2753497.2753503.

[46] M. Youssef and A. Agrawala, “The Horus location determination system,” *Wireless Networks*, vol. 14, no. 3, pp. 357–374, Jun. 2008, doi: 10.1007/s11276-006-0725-7.

[47] S. Shi, S. Sigg, L. Chen, and Y. Ji, “Accurate Location Tracking from CSI-Based Passive Device-Free Probabilistic Fingerprinting,” *IEEE Trans Veh Technol*, vol. 67, no. 6, pp. 5217–5230, Jun. 2018, doi: 10.1109/TVT.2018.2810307.

[48] D. J. Suroso, N. A. Siddiq, R. Rupaksi, and A. B. Krisnawan, “Low-cost and Simple Configuration Device-free Indoor Localization: A Review,” in *Proceedings - 2020 6th International Conference on Science and Technology, ICST 2020*, Institute of Electrical and Electronics Engineers Inc., 2020. doi: 10.1109/ICST50505.2020.9732779.

[49] J. Wang and J. G. Park, “An enhanced indoor positioning algorithm based on fingerprint using fine-grained csi and rssi measurements of ieee 802.11n wlan,” *Sensors*, vol. 21, no. 8, Apr. 2021, doi: 10.3390/s21082769.

[50] L. Zhao, H. Wang, P. Li, and J. Liu, “An Improved WiFi Indoor Localization Method Combining Channel State Information and Received Signal Strength,” in *36th Chinese Control Conference*, Dalian, Jul. 2017, pp. 8964–8969.

[51] Z. Hu, G. Huang, Y. Hu, and Z. Yang, “Wi-Vi fingerprint: WiFi and vision integrated fingerprint for smartphone-based indoor self-localization,” in *ICIP 2017*, 2017, pp. 4402–4406.

[52] G. Huang, Z. Hu, J. Wu, H. Xiao, and F. Zhang, “WiFi and Vision-Integrated Fingerprint for Smartphone-Based Self-Localization in Public Indoor Scenes,” *IEEE Internet Things J*, vol. 7, no. 8, pp. 6748–6761, Aug. 2020, doi: 10.1109/JIOT.2020.2974928.

[53] G. Chandra, “Position Estimation at Indoors using Wi-Fi and Magnetic Field Sensors,” in *IPIN 2022 WiP Proceeding*, 2022. [Online]. Available: http://ceur-ws.org

[54] A. Tsanousa, V. R. Xefteris, G. Meditskos, S. Vrochidis, and I. Kompatsiaris, “Combining rssi and accelerometer features for room-level localization,” *Sensors*, vol. 21, no. 8, Apr. 2021, doi: 10.3390/s21082723.

[55] W. Ruan, Q. Z. Sheng, L. Yao, T. Gu, M. Ruta, and L. Shangguan, “Device-free indoor localization and tracking through Human-Object Interactions,” in *WoWMoM 2016 - 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Institute of Electrical and Electronics Engineers Inc., Jul. 2016. doi: 10.1109/WoWMoM.2016.7523524.

[56] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu, and A. Chen, "Reducing fingerprint collection for indoor localization," *Comput Commun*, vol. 83, pp. 56–63, Jun. 2016, doi: 10.1016/j.comcom.2015.09.022.

[57] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals," *IEEE Trans Veh Technol*, vol. 64, no. 4, pp. 1340–1353, Apr. 2015, doi: 10.1109/TVT.2015.2397598.

[58] V. Moghtadaiee, S. A. Ghorashi, and M. Ghavami, "New Reconstructed Database for Cost Reduction in Indoor Fingerprinting Localization," *IEEE Access*, vol. 7, pp. 104462–104477, 2019, doi: 10.1109/ACCESS.2019.2932024.

[59] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Structured Group Sparsity: A Novel Indoor WLAN Localization, Outlier Detection, and Radio Map Interpolation Scheme," *IEEE Trans Veh Technol*, vol. 66, no. 7, pp. 6498–6510, Jul. 2017, doi: 10.1109/TVT.2016.2631980.

[60] A. Zayets and E. Steinbach, "Interpolation and extrapolation of multipath fingerprints using virtual transmitter placement," in *IEEE International Conference on Communications*, Institute of Electrical and Electronics Engineers Inc., Jul. 2018. doi: 10.1109/ICC.2018.8422206.

[61] Y. Li, G. Shi, X. Zhou, W. Qu, and K. Li, "Reducing the site survey using fingerprint refinement for cost-efficient indoor location," *Wireless Networks*, vol. 25, no. 3, pp. 1201–1213, Apr. 2019, doi: 10.1007/s11276-018-1711-6.

[62] J. Bi, Y. Wang, H. Cao, H. Qi, K. Liu, and Xu. Shenglei, "A Method of Radio Map Construction based on Crowdsourcing and Interpolation for Wi-Fi Positioning System," in *PIN 2018 : Ninth International Conference on Indoor Positioning and Indoor Navigation*, Nantes, Sep. 2018.

[63] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," 2016.

[64] A. I. Miller, "Ian Goodfellow's Generative Adversarial Networks: AI Learns to Imagine," *The Artist in the Machine*, 2020, doi: 10.7551/mitpress/11585.003.0015.

[65] I. Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Networks," 2016.

[66] J. Langr and V. Bok, *Deep learning with Generative Adversarial Networks: GANs in Action*. NewYork, 2019.

[67] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, and B. A. Research, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks Monet Photos."

[68] J. M. Tomczak, “Deep Generative Modeling,” 2022. doi: https://doi.org/10.1007/978-3-030-93158-2.

[69] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, no. Ml, pp. 1–14, 2014.

[70] L. Weng, “From Autoencoder to Beta-VAE,” *lilianweng.github.io*, 2018.

[71] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science (1979)*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[72] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” Dec. 2013, [Online]. Available: http://arxiv.org/abs/1312.6114

[73] D. P. Kingma Google, M. Welling, and B. -Delft, “An Introduction to Variational Autoencoders,” *Foundations and Trends R in Machine Learning*, vol. xx, No. xx, pp. 1–18, 2019, doi: 10.1561/XXXXXXXXX.

[74] I. J. Goodfellow *et al.*, “Generative Adversarial Nets.” [Online]. Available: http://www.github.com/goodfeli/adversarial

[75] S. Hitawala and D. R. Cheriton, “Comparative Study on Generative Adversarial Networks,” 2018.

[76] M. Nabati, H. Navidan, R. Shahbazian, S. A. Ghorashi, and D. Windridge, “Using Synthetic Data to Enhance the Accuracy of Fingerprint-Based Localization: A Deep Learning Approach,” *IEEE Sens Lett*, vol. 4, no. 4, Apr. 2020, doi: 10.1109/LSENS.2020.2971555.

[77] W. Njima, M. Chafii, A. Chorti, R. M. Shubair, and H. V. Poor, “Indoor Localization Using Data Augmentation via Selective Generative Adversarial Networks,” *IEEE Access*, vol. 9, pp. 98337–98347, 2021, doi: 10.1109/ACCESS.2021.3095546.

[78] L. Chen, S. Zhang, H. Tan, and B. Lv, “Progressive RSS Data Augmenter with Conditional Adversarial Networks,” *IEEE Access*, vol. 8, pp. 26975–26983, 2020, doi: 10.1109/ACCESS.2020.2971269.

[79] H. Rizk, A. Shokry, and M. Youssef, “Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning,” in *IEEE Wireless Communications and Networking Conference, WCNC*, Institute of Electrical and Electronics Engineers Inc., Apr. 2019. doi: 10.1109/WCNC.2019.8886005.

[80] K. M. Chen and R. Y. Chang, “A Comparative Study of Deep-Learning-Based Semi-Supervised Device-Free Indoor Localization,” in *2021 IEEE Global Communications Conference, GLOBECOM 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/GLOBECOM46510.2021.9685548.

[81] W. Qian, F. Lauri, and F. Gechter, “Supervised and semi-supervised deep probabilistic models for indoor positioning problems,” *Neurocomputing*, vol. 435, pp. 228–238, May 2021, doi: 10.1016/j.neucom.2020.12.131.

[82] Z. Li, J. Cao, H. Wang, and M. Zhao, “Sparsely Self-Supervised Generative Adversarial Nets for Radio Frequency Estimation,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2428–2442, Nov. 2019, doi: 10.1109/JSAC.2019.2933779.

[83] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, “Fidora: Robust WiFi-Based Indoor Localization via Unsupervised Domain Adaptation,” *IEEE Internet Things J*, vol. 9, no. 12, pp. 9872–9888, Jun. 2022, doi: 10.1109/JIOT.2022.3163391.

[84] F. Alhomayani and M. H. Mahoor, “Oversampling Highly Imbalanced Indoor Positioning Data using Deep Generative Models,” Aug. 2021, [Online]. Available: http://arxiv.org/abs/2108.13503

[85] A. Zhang, K. Zhu, R. Wang, and C. Yi, “Missing data inference for crowdsourced radio map construction: An adversarial auto-encoder method,” in *IEEE Wireless Communications and Networking Conference, WCNC*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/WCNC49053.2021.9417253.

[86] H. Zou *et al.*, “Adversarial Learning-Enabled Automatic WiFi Indoor Radio Map Construction and Adaptation with Mobile Robot,” *IEEE Internet Things J*, vol. 7, no. 8, pp. 6946–6954, Aug. 2020, doi: 10.1109/JIOT.2020.2979413.

[87] W. Qian, F. Lauri, and F. Gechter, “Supervised and semi-supervised deep probabilistic models for indoor positioning problems,” *Neurocomputing*, vol. 435, pp. 228–238, May 2021, doi: 10.1016/j.neucom.2020.12.131.

[88] J. Wang, Q. Gao, Y. Yu, P. Cheng, L. Wu, and H. Wang, “Robust device-free wireless localization based on differential RSS measurements,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5943–5952, 2013, doi: 10.1109/TIE.2012.2228145.

[89] X. Dang, X. Si, Z. Hao, and Y. Huang, “A novel passive indoor localization method by fusion csi amplitude and phase information,” *Sensors (Switzerland)*, vol. 19, no. 4, Feb. 2019, doi: 10.3390/s19040875.

[90] S. Palipana, B. Pietropaoli, and D. Pesch, “Recent advances in RF-based passive device-free localisation for indoor applications,” *Ad Hoc Networks*, vol. 64, pp. 80–98, Sep. 2017, doi: 10.1016/j.adhoc.2017.06.007.

[91] Y. Guo, K. Huang, N. Jiang, X. Guo, Y. Li, and G. Wang, “An exponential-rayleigh model for RSS-based device-free localization and tracking,” *IEEE Trans Mob Comput*, vol. 14, no. 3, pp. 484–494, Mar. 2015, doi: 10.1109/TMC.2014.2329007.

[92] L. Zhang, Q. Gao, X. Ma, J. Wang, T. Yang, and H. Wang, “Defi: Robust training-free device-free wireless localization with wifi,” *IEEE Trans Veh Technol*, vol. 67, no. 9, pp. 8822–8831, Sep. 2018, doi: 10.1109/TVT.2018.2850842.

[93] S. Denis, R. Berkvens, and M. Weyn, “A survey on detection, tracking and identification in radio frequency-based device-free localization,” *Sensors (Switzerland)*, vol. 19, no. 23. MDPI AG, Dec. 01, 2019. doi: 10.3390/s19235329.

[94] W. Ruan, L. Yao, Q. Z. Sheng, N. J. G. Falkner, and X. Li, “TagTrack: Device-Free Localization and Tracking Using Passive RFID Tags,” in *11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ICST, Nov. 2014. doi: http://dx.doi.org/10.4108/icst.mobiquitous.2014.258004.

[95] W. Ruan, Q. Z. Sheng, L. Yao, T. Gu, M. Ruta, and L. Shangguan, “Device-free indoor localization and tracking through Human-Object Interactions,” in *WoWMoM 2016 - 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Institute of Electrical and Electronics Engineers Inc., Jul. 2016. doi: 10.1109/WoWMoM.2016.7523524.

[96] M. Scherhäufl, M. Pichler, E. Schimbäck, D. J. Müller, A. Ziroff, and A. Stelzer, “Indoor localization of passive UHF RFID tags based on phase-of-arrival evaluation,” *IEEE Trans Microw Theory Tech*, vol. 61, no. 12, pp. 4724–4729, Dec. 2013, doi: 10.1109/TMTT.2013.2287183.

[97] J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica, “Performance evaluation of bluetooth low energy: A systematic review,” *Sensors (Switzerland)*, vol. 17, no. 12. MDPI AG, Dec. 13, 2017. doi: 10.3390/s17122898.

[98] J. R. Jiang, H. Subakti, and H. S. Liang, “Fingerprint feature extraction for indoor localization†,” *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165434.

[99] J. Zheng, C. Wu, H. Chu, and Y. Xu, "An improved RSSI measurement in wireless sensor networks," in *Procedia Engineering*, 2011, pp. 876–880. doi: 10.1016/j.proeng.2011.08.162.

[100] A. S. A. Sukor, L. M. Kamarudin, A. Zakaria, N. A. Rahim, S. Sudin, and H. Nishizaki, "Rssi-based for device-free localization using deep learning technique," *Smart Cities*, vol. 3, no. 2, Jun. 2020, doi: 10.3390/smartcities3020024.

[101] S. Dolha, P. Negirla, F. Alexa, and I. Silea, "Considerations about the signal level measurement in wireless sensor networks for node position estimation," *Sensors (Switzerland)*, vol. 19, no. 19, Oct. 2019, doi: 10.3390/s19194179.

[102] D. J. Suroso, A. S. H. Rudianto, M. Arifin, and S. Hawibowo, "Random forest and interpolation techniques for fingerprint-based indoor positioning system in un-ideal Environment," *International Journal of Computing and Digital Systems*, vol. 10, no. 1, 2021, doi: 10.12785/IJCDS/100166.

[103] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent Neural Networks for Accurate RSSI Indoor Localization," *IEEE Internet Things J*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019, doi: 10.1109/JIOT.2019.2940368.

[104] T. K. Geok *et al.*, "Review of indoor positioning: Radio wave technology," *Applied Sciences (Switzerland)*, vol. 11, no. 1. MDPI AG, pp. 1–44, Jan. 01, 2021. doi: 10.3390/app11010279.

[105] Z. Wu, Q. Xu, J. Li, C. Fu, Q. Xuan, and Y. Xiang, "Passive Indoor Localization Based on CSI and Naive Bayes Classification," *IEEE Trans Syst Man Cybern Syst*, vol. 48, no. 9, pp. 1566–1577, Sep. 2018, doi: 10.1109/TSMC.2017.2679725.

[106] B. Jang and H. Kim, "Indoor positioning technologies without offline fingerprinting map: A survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 508–525, 2019, doi: 10.1109/COMST.2018.2867935.

[107] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *IEEE Trans Mob Comput*, vol. 12, no. 7, pp. 1321–1334, 2013, doi: 10.1109/TMC.2012.106.

[108] H. Wu, X. Ma, C. H. H. Yang, and S. Liu, "Convolutional Neural Network Based Radio Tomographic Imaging," in *2020 54th Annual Conference on Information Sciences and Systems, CISS 2020*, Institute of Electrical and Electronics Engineers Inc., Mar. 2020. doi: 10.1109/CISS48834.2020.1570617238.

[109] J. Wilson and N. Patwari, “Radio tomographic imaging with wireless networks,” *IEEE Trans Mob Comput*, vol. 9, no. 5, pp. 621–632, May 2010, doi: 10.1109/TMC.2009.174.

[110] V. Nguyen, M. Ibrahim, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard, “EyeLight: Light-and-shadow-based Occupancy Estimation and Room Activity Recognition.”

[111] Z. Zhao *et al.*, “NaviLight: Indoor localization and navigation under arbitrary lights,” in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, IEEE, May 2017. doi: 10.1109/INFOCOM.2017.8057184.

[112] Y. Yang, J. Hao, J. Luo, and S. J. Pan, “CeilingSee: Device-free occupancy inference through lighting infrastructure based LED sensing,” in *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, Institute of Electrical and Electronics Engineers Inc., May 2017, pp. 247–256. doi: 10.1109/PERCOM.2017.7917871.

[113] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, “Wireless RSSI fingerprinting localization,” *Signal Processing*, vol. 131. Elsevier B.V., pp. 235–244, Feb. 01, 2017. doi: 10.1016/j.sigpro.2016.07.005.

[114] A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr, “An enhanced WiFi indoor localization System based on machine learning,” in *2016 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016*, Institute of Electrical and Electronics Engineers Inc., Nov. 2016. doi: 10.1109/IPIN.2016.7743586.

[115] G. Louppe, “Understanding Random Forest: From Theory to Practice,” University in Liège, Liège, 2014. doi: https://doi.org/10.48550/arXiv.1407.7502.

[116] D. Han, S. hoon Jung, and S. Lee, “A sensor fusion method for Wi-Fi-based indoor positioning,” *ICT Express*, vol. 2, no. 2, pp. 71–74, Jun. 2016, doi: 10.1016/j.icte.2016.04.002.

[117] X. Guo, N. Ansari, L. Li, and H. Li, “Indoor Localization by Fusing a Group of Fingerprints Based on Random Forests,” *IEEE Internet Things J*, vol. 5, no. 6, pp. 4686–4698, Dec. 2018, doi: 10.1109/JIOT.2018.2810601.

[118] S. Palipana, B. Pietropaoli, and D. Pesch, “Recent advances in RF-based passive device-free localisation for indoor applications,” *Ad Hoc Networks*, vol. 64, pp. 80–98, 2017, doi: 10.1016/j.adhoc.2017.06.007.

[119] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, "A comparative study on machine learning algorithms for indoor positioning," *INISTA 2015 - 2015 International Symposium on Innovations in Intelligent SysTems and Applications, Proceedings*, 2015, doi: 10.1109/INISTA.2015.7276725.

[120] L. Breiman, "Random Forests," 2001.

[121] S. Lee, J. Kim, and N. Moon, "Random forest and WiFi fingerprint-based indoor location recognition system using smart watch," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, Dec. 2019, doi: 10.1186/s13673-019-0168-7.

[122] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Comput Sci Eng*, vol. 13, no. 2, pp. 22–30, 2011, doi: 10.1109/MCSE.2011.37.

[123] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput Sci Eng*, vol. 9, no. 3, pp. 99–104, 2007, doi: 10.1109/MCSE.2007.55.

[124] R. Zhou, M. Hao, X. Lu, M. Tang, and Y. Fu, "Device-free localization based on CSI fingerprints and deep neural networks," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2018*, Institute of Electrical and Electronics Engineers Inc., Jun. 2018, pp. 1–9. doi: 10.1109/SAHCN.2018.8397121.

[125] M. Marjani *et al.*, "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017, doi: 10.1109/ACCESS.2017.2689040.

[126] S. Bartoletti, A. Conti, D. Dardari, and A. Giorgetti, "5G Localization and Context-Awareness," 2018.

[127] G. Xu, *GPS · Theory, Algorithms and Applications*, 2nd ed. Springer, 2007.

[128] C. Yang and H.-R. Shao, "WiFi-based Indoor Positioning," *IEEE Communication Magazine*, 2015.

[129] D. Yan, B. Kang, H. Zhong, and R. Wang, "Research on Positioning System based on ZigBee Communication," in *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC 2018) : October 12-14, 2018, Chongqing, China*, Chongqing: IEEE, Oct. 2018.

[130] R. Mehta, J. Sahni, and K. Khanna, "Internet of Things: Vision, Applications and Challenges," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 1263–1269. doi: 10.1016/j.procs.2018.05.042.

[131] M. Carlos-Mancilla, E. López-Mellado, and M. Siller, “Wireless sensor networks formation: Approaches and techniques,” *Journal of Sensors*, vol. 2016. Hindawi Limited, 2016. doi: 10.1155/2016/2081902.

[132] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes: Cooperative localization in wireless sensor networks,” *IEEE Signal Process Mag*, vol. 22, no. 4, pp. 54–69, 2005, doi: 10.1109/MSP.2005.1458287.

[133] K. Srinivasan and P. Levis, “RSSI is Under Appreciated,” in *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*, 2006.

[134] K. Heurtefeux and F. Valois, “Is RSSI a good choice for localization in wireless sensor network?,” in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2012, pp. 732–739. doi: 10.1109/AINA.2012.19.

[135] A. T. Parameswaran, M. I. Husain, and S. Upadhyaya, “Is RSSI a Reliable Parameter in Sensor Localization Algorithms-An Experimental Study,”

[136] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, “Indoor positioning algorithm based on the improved rssi distance model,” *Sensors (Switzerland)*, vol. 18, no. 9, Sep. 2018, doi: 10.3390/s18092820.

[137] Q. D. Vo and P. De, “A survey of fingerprint-based outdoor localization,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1. Institute of Electrical and Electronics Engineers Inc., pp. 491–506, Jan. 01, 2016. doi: 10.1109/COMST.2015.2448632.

[138] E. Hatem, S. Fortes, E. Colin, S. Abou-Chakra, J. M. Laheurte, and B. El-Hassan, “Accurate and low-complexity auto-fingerprinting for enhanced reliability of indoor localization systems,” *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165346.

[139] W. Ji, K. Zhao, Z. Zheng, C. Yu, and S. Huang, “Multivariable Fingerprints with Random Forest Variable Selection for Indoor Positioning System,” *IEEE Sens J*, vol. 22, no. 6, pp. 5398–5406, Mar. 2022, doi: 10.1109/JSEN.2021.3103863.

[140] L. Wang, S. Tiku, and S. Pasricha, “CHISEL: Compression-Aware High-Accuracy Embedded Indoor Localization With Deep Learning,” *IEEE Embed Syst Lett*, vol. 14, no. 1, pp. 23–26, Mar. 2022, doi: 10.1109/LES.2021.3094965.

[141] G. Chen, X. Meng, Y. Wang, Y. Zhang, P. Tian, and H. Yang, “Integrated WiFi/PDR/smartphone using an unscented Kalman filter algorithm for 3D indoor localization,” *Sensors (Switzerland)*, vol. 15, no. 9, pp. 24595–24614, Sep. 2015, doi: 10.3390/s150924595.

[142] J. Yan, G. Qi, B. Kang, X. Wu, and H. Liu, “Extreme Learning Machine for Accurate Indoor Localization Using RSSI Fingerprints in Multifloor Environments,” *IEEE Internet Things J*, vol. 8, no. 19, pp. 14623–14637, Oct. 2021, doi: 10.1109/JIOT.2021.3071152.

[143] J. Y. Zhu, A. X. Zheng, J. Xu, and V. O. K. Li, “Spatio-temporal (S-T) similarity model for constructing WIFI-based RSSI fingerprinting map for indoor localization,” in *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, Institute of Electrical and Electronics Engineers Inc., 2014, pp. 678–684. doi: 10.1109/IPIN.2014.7275543.

[144] Y. Sun, Y. He, and Y. Yang, “Interpolation Method for Radio Map Establishment Based on RSS Clustering and Propagation Model Optimization,” in *Proceedings - 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2018*, Institute of Electrical and Electronics Engineers Inc., Feb. 2019, pp. 451–454. doi: 10.1109/CyberC.2018.00087.

[145] Y. Assayag, H. Oliveira, E. Souto, R. Barreto, and R. Pazzi, “Indoor Positioning System Using Synthetic Training and Data Fusion,” *IEEE Access*, vol. 9, pp. 115687–115699, 2021, doi: 10.1109/ACCESS.2021.3105188.

[146] F. Y. M. Adiyatma, A. E. Kurniawan, D. J. Suroso, and P. Cherntanomwong, “Performance Comparison of Several Range- based Techniques for Indoor Localization Based on Received Signal Strength Indicator,” vol. 7, no. 1, pp. 40–53, 2021, doi: 10.34818/ijoict.v7i1.550.

[147] S. Djosic, I. Stojanovic, M. Jovanovic, T. Nikolic, and G. L. Djordjevic, “Fingerprinting-assisted UWB-based localization technique for complex indoor environments,” *Expert Syst Appl*, vol. 167, Apr. 2021, doi: 10.1016/j.eswa.2020.114188.

[148] L. Zhang and H. Wang, “3D-WiFi: 3D Localization with Commodity WiFi,” *IEEE Sens J*, vol. 19, no. 13, pp. 5141–5152, Jul. 2019, doi: 10.1109/JSEN.2019.2900511.

[149] W. Kui, S. Mao, X. Hei, and F. Li, “Towards Accurate Indoor Localization Using Channel State Information,” in *2018 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2018*, Institute of Electrical and Electronics Engineers Inc., Aug. 2018. doi: 10.1109/ICCE-China.2018.8448497.

[150] X. Wang, Z. Chen, S. Zhang, and J. Zhu, “Super-Resolution Based Fingerprint Augment for Indoor WiFi Localization,” in *2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/GLOBECOM42002.2020.9348146.

[151] Y.-K. Chen, H.-J. Chou, and R. Y. Chang, "Machine-Learning Indoor Localization with Access Point Selection and Signal Strength Reconstruction," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring) : proceedings : Nanjing, China, 15-18 May 2016.*, Institute of Electrical and Electronics Engineers, 2016.

[152] P. Cherntanomwong and D. J. Suroso, "Indoor Localization System using Wireless Sensor Networks for Stationary and Moving Target," in *2011 8th International Conference on Information, Communications and Signal Processing.*, IEEE, 2011.

[153] W. H. Press, *Numerical recipes in C : the art of scientific computing*. Cambridge University Press, 1992.

[154] J. Kiusalaas, *Numerical Methods in Engineering with Python 3*. Cambridge University Press, 2013. doi: 10.1017/CBO9781139523899.

[155] E. Ostertagová, "Modelling using polynomial regression," in *Procedia Engineering*, Elsevier Ltd, 2012, pp. 500–506. doi: 10.1016/j.proeng.2012.09.545.

[156] D. J. Suroso, P. Cherntanomwong, P. Sooraksa, and J. Takada, "Location fingerprint technique using Fuzzy C-Means clustering algorithm for indoor localization," in *TENCON 2011 - 2011 IEEE Region 10 Conference*, IEEE, Nov. 2011, pp. 88–92. doi: 10.1109/TENCON.2011.6129069.

[157] D. J. Suroso, P. Cherntanomwong, P. Sooraksa, and J.-I. Takada, "Fingerprint-based technique for indoor localization in wireless sensor networks using Fuzzy C-Means clustering algorithm," in *2011 International Symposium on Intelligent Signal Processing and Communications Systems: "The Decade of Intelligent and Green Signal Processing and Communications", ISPACS 2011*, 2011. doi: 10.1109/ISPACS.2011.6146167.

[158] P. Cherntanomwong, J. I. Takada, and H. Tsuji, "Signal subspace interpolation from discrete measurement samples In constructing a database for location fingerprint technique," *IEICE Transactions on Communications*, vol. E92-B, no. 9, pp. 2922–2930, 2009, doi: 10.1587/transcom.E92.B.2922.

[159] A. Manirabona and L. C. Fourati, "A Kriged Fingerprinting for Wireless Body Area Network Indoor Localization," *Wireless Personal Communications*, vol. 80, no. 4. Kluwer Academic Publishers, pp. 1501–1515, Feb. 01, 2015. doi: 10.1007/s11277-014-2095-2.

[160] J. Bi, Y. Wang, H. Cao, H. Qi, K. Liu, and S. Xu, "A Method of Radio Map Construction Based on Crowdsourcing and Interpolation for Wi-Fi Positioning System," *IPIN 2018*

- *9th International Conference on Indoor Positioning and Indoor Navigation*, no. September, pp. 24–27, 2018, doi: 10.1109/IPIN.2018.8533749.

[161] Y. Li, G. Shi, X. Zhou, W. Qu, and K. Li, "Reducing the site survey using fingerprint refinement for cost-efficient indoor location," *Wireless Networks*, vol. 25, no. 3, pp. 1201–1213, 2019, doi: 10.1007/s11276-018-1711-6.

[162] T. King, T. Haenselmann, and W. Effelsberg, "On-demand fingerprint selection for 802.11-based positioning systems," *2008 IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM2008*, 2008, doi: 10.1109/WOWMOM.2008.4594839.

[163] J. Y. Zhu, A. X. Zheng, J. Xu, and V. O. K. Li, "Spatio-temporal (S-T) similarity model for constructing WIFI-based RSSI fingerprinting map for indoor localization," *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, no. October, pp. 678–684, 2014, doi: 10.1109/IPIN.2014.7275543.

[164] Y. Sun, Y. He, and Y. Yang, "Interpolation Method for Radio Map Establishment Based on RSS Clustering and Propagation Model Optimization," *Proceedings - 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2018*, pp. 451–454, 2019, doi: 10.1109/CyberC.2018.00087.

[165] S. Tiku, P. Kale, and S. Pasricha, "QuickLoc: Adaptive deep-learning for fast indoor localization with mobile devices," *ACM Transactions on Cyber-Physical Systems*, vol. 5, no. 4, pp. 1–28, 2021, doi: 10.1145/3461342.

[166] D. J. Suroso, P. Cherntanomwong, and P. Sooraksa, "Synthesis of a Small Fingerprint Database through a Deep Generative Model for Indoor Localisation," *Elektronika ir Elektrotechnika*, vol. 29, no. 1, pp. 69–75, Feb. 2023, doi: 10.5755/j02.eie.31905.

[167] A. Basiri *et al.*, "Indoor location based services challenges, requirements and usability of current solutions," *Comput Sci Rev*, vol. 24, pp. 1–12, May 2017, doi: 10.1016/j.cosrev.2017.03.002.

[168] P. Crane, Z. Huang, and H. Zhang, "CRAFT reducing the effort for indoor localisation," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, vol. 2017-Octob, pp. 1–5, 2018, doi: 10.1109/PIMRC.2017.8292211.

[169] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 6, Nov. 2021, doi: 10.1007/s42979-021-00815-1.

[170] R. S. Sinha, S. M. Lee, M. Rim, and S. H. Hwang, "Data augmentation schemes for deep learning in an indoor positioning application," *Electronics (Switzerland)*, vol. 8, no. 5, May 2019, doi: 10.3390/electronics8050554.

[171] X. Feng, K. A. Nguyen, and Z. Luo, "A survey of deep learning approaches for WiFi-based indoor positioning," *Journal of Information and Telecommunication*, vol. 6, no. 2, pp. 163–216, 2022, doi: 10.1080/24751839.2021.1975425.

[172] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1–9. [Online]. Available: http://www.github.com/goodfeli/adversarial

[173] A. Haider, Y. Wei, S. Liu, and S. H. Hwang, "Pre- and post-processing algorithms with deep learning classifier for Wi-Fi fingerprint-based indoor positioning," *Electronics (Switzerland)*, vol. 8, no. 2, Feb. 2019, doi: 10.3390/electronics8020195.

[174] M. Karmy, S. Elsayed, and A. Zekry, "Performance enhancement of an indoor localization system based on visible light communication using rssi/tdoa hybrid technique," *Journal of Communications*, vol. 15, no. 5, pp. 379–389, May 2020, doi: 10.12720/jcm.15.5.379-389.

[175] X. Ding, L. Gao, and Z. Wang, "Modified fingerprinting algorithm for indoor location," *Journal of Communications*, vol. 12, no. 3, pp. 145–151, Mar. 2017, doi: 10.12720/jcm.12.3.145-151.

[176] X. Song *et al.*, "Cnnloc: Deep-learning based indoor localization with wifi fingerprinting," in *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, Institute of Electrical and Electronics Engineers Inc., Aug. 2019, pp. 589–595. doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00139.

[177] R. S. Sinha and S. H. Hwang, "Comparison of CNN applications for rssi-based fingerprint indoor localization," *Electronics (Switzerland)*, vol. 8, no. 9, Sep. 2019, doi: 10.3390/electronics8090989.

[178] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[179] S. Hitawala, "Comparative Study on Generative Adversarial Networks," Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.04271

[180] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[181] 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 (Switzerland)*, vol. 15, no. 1, pp. 715–732, Jan. 2015, doi: 10.3390/s150100715.

[182] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, “Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1443–1457, Jul. 2015, doi: 10.1109/JSAC.2015.2430274.

[183] R. Zhou, M. Hao, X. Lu, M. Tang, and Y. Fu, “Device-free localization based on CSI fingerprints and deep neural networks,” *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2018*, pp. 1–9, 2018, doi: 10.1109/SAHCN.2018.8397121.

Appendices

Publication List

Journal:

1. **D. J. Suroso**, P. Cherntanomwong, and P. Sooraksa, "Indoor Device-free Localization Using Received Signal Strength Indicator and Illuminance Sensor for Random-forest-based Fingerprint Technique," Sensors and Materials, vol. 33, no. 12, pp. 4331–4345, December 2021, doi: 10.18494/SAM.2021.3632. (Q3, Scopus and Web of Science)
2. **D. J. Suroso**, F. Y. M. Adiyatma, P. Cherntanomwong, and P. Sooraksa, "Fingerprint Database Enhancement by Applying Interpolation and Regression Techniques for IoT-based Indoor Localization," Emerging Science Journal, vol. 4, pp. 167–189, January 2022, doi: 10.28991/esj-2021-sp1-012. (Q1, Scopus)
3. **D. J. Suroso**, P. Cherntanomwong, and P. Sooraksa, "Synthesis of a Small Fingerprint Database through a Deep Generative Model for Indoor Localisation," Elektronika Ir Elektrotechnika, vol. 29 no. 1, pp. 69–75, February 2023, doi: 10.5755/j02.eie.31905. (Q3, Scopus and Web of Science)

Conference Proceeding:

1. **D. J. Suroso**, P. Cherntanomwong, and P. Sooraksa, "Deep Generative Model-based RSSI Synthesis for Indoor Localization," 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON), Prachuap Khiri Khan, Thailand, 2022, pp. 1-5, doi: 10.1109/ECTI-CON54298.2022.9795409. (IEEE Explore, Scopus)
2. **D. J. Suroso**, P. Sooraksa, and P. Cherntanomwong, "Is Deep Diffusion Probabilistic Model Applicable for Fingerprint-based Indoor Localization?" 2022 26th International Computer Science and Engineering Conference (ICSEC), Sakon Nakhon, Thailand, 2022, pp. 202-207, doi: 10.1109/ICSEC56337.2022.10049366. (IEEE Explore, Scopus).

3. F. Y. M. Adiyatma, **D. J. Suroso** and P. Cherntanomwong, "Fingerprint Database Enhancement using Spatial Interpolation for IoT-based Indoor Localization," *2022 26th International Computer Science and Engineering Conference (ICSEC)*, Sakon Nakhon, Thailand, 2022, pp. 192-197, doi: 10.1109/ICSEC56337.2022.10049367. (IEEE Explore, Scopus)

4. **D. J. Suroso**, F. Y. M. Adiyatma, P. Cherntanomwong, and P. Sooraksa, "Implementation of Wi-Fi-based Indoor Positioning System: Challenges and Future Possibilities," The 5th International Conference on Computational Intelligence in Information Systems (Online Conference), October 2022. (AIP, Scopus), *to be published*.

5. **D. J. Suroso**, P. Cherntanomwong, and P. Sooraksa, "Fingerprint-based Indoor Localization via Deep Learning," The 6th International Conference on Electronics, Communications and Control Engineering (ICECC) 2023, Fukuoka Institute of Technology, Fukuoka, Japan on March 24-26, 2023. (ACM, Scopus), *to be published*.

6. F. Y. M. Adiyatma, **D. J. Suroso** and P. Cherntanomwong, "Regression-based Path Loss Model Correction to Construct Fingerprint Database for Indoor Localization," The 6th International Conference on Electronics, Communications and Control Engineering (ICECC) 2023, Fukuoka Institute of Technology, Fukuoka, Japan on March 24-26, 2023. (ACM, Scopus), *to be published*.

7. **D. J. Suroso**, A. B. Krisnawan, P. Sooraksa, and P. Cherntanomwong, "3D Range-Based Indoor Localization by Using Only Two Beacons," 2021 3rd Engineering Physics International Conference (EPIC), published 23 May 2023.

Achievement

Best Presentation Award at The 6th International Conference on Electronics, Communications and Control Engineering (ICECC) 2023, Fukuoka Institute of Technology, Fukuoka, Japan on March 24-26, 2023.

GitHub Repository

https://github.com/dwijokosuroso

Author Biography

Full Name	Dwi Joko Suroso
Date, birthplace	23 August 1988 in Sleman, Indonesia
Address	Krapyak Triharjo 10/11 Sleman, Sleman, Yogyakarta 55514 Indonesia Telp. +62-8123-22-93-900 (Office: Dept. of Nuclear Engineering and Engineering Physics, Faculty of Engineering, Universitas Gadjah Mada (UGM), Jl. Grafika No. 2, Yogyakarta 55281, Indonesia)
Education	2012 Master of Engineering in Computer Engineering King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand 2010 Bachelor of Engineering in Engineering Physics Universitas Gadjah Mada (UGM), Indonesia
Area of Expertise.	1.) Wireless Communication 2.) Machine/Deep Learning 3.) Instrumentation (Automation)
Work Experience	
2017 - present	Lecturer at Dept. Nuclear Engineering and Engineering Physics, Universitas Gadjah Mada (UGM), Indonesia.
Honorary Awards	
2023	Best Presentation Award at ICECC 2023, Fukuoka Institute of Technology, Japan.
2016	Best Presentation Award at Interdisciplinary Workshop, Tokyo Institute of Technology, Japan.
2011	High Quality Paper Award ICICS 2011, IEEE Singapore.