# Deep Learning-based Indoor Localization for IoT Applications

**by Qianwen Ye**

Thesis submitted in fulfilment of the requirements for the degree of

**Doctor of Philosophy**

under the supervision of Professor Gengfa Fang

University of Technology Sydney
Faculty of Engineering and Information Technology

May 2023

# Certification of Original Authorship

I, Qianwen Ye, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, at the University of Technology Sydney (UTS).

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with Beijing University of Posts and Communications (BUPT).

This research is supported by the Australian Government Research Training Program.

Signature:____________

Date: 25.05.2023

# Acknowledgements

I have had an unforgettable and life-changing experience at University of Technology Sydney. Firstly, I want to salute my principal supervisor, Professor Gengfa Fang because of his full supervision, encourage guidance, and earnest teaching. Professor Fang, always provide me with exciting and constructive suggestions in research and helped me on my lives in Australia. His enthusiasm, carefulness, and ability related to research, guidance, and managing have enlightened me to strive to be a better version of myself. Above all, he is an extremely strong, sociable and enthusiastic person, and I am deeply grateful to him for being my teacher, and it is my great honor to conduct research under his supervision. I would like to thank Eryk Dutkiewicz, Forest Zhu and Jack Wang, three of my co-supervisors because they have been constructive and original in guiding my research and study.

As a Dual-PhD student between UTS and Beijing University of Posts and Communications (BUPT), I have also studied at BUPT. I want to pay my respect to my supervisor at BUPT, Professor Hongxia Bie due to her sincere supervision, constant encouragement, and firmly support. At BUPT, her support and supervision on my research are critical factors for my success and completion. Even when I was at UTS, she ran regular consultation meetings with me remotely. Apart from research study, she has also been a mentor to me on my life. She is really a gentle, responsible, and professional woman.

I would also like to thank my collaborators: Xiaochen Fan and Xudong Song. Xiaochen Fan is a cautious and patient researcher focusing on computer science and technology. We collaborated deeply on my papers on deep learning-based indoor localization. His patience and carefulness prompt me to be a more productive researcher. Xudong Song is a talented fresh Ph.D. candidate and an excellent programmer focusing on deep learning. We exchanged a lot of deep learning-based

knowledge with each other, which inspired me a lot on my research.

I would also like to pay a tribute to my parents Zhiqing Ye and Jiaohua Chen, for their encouragement and support. My father works in a junior high school as a math teacher. His serious and responsible attitude to his work and students encouraged me to have a serious attitude to my research and study. My mother has done a lot for the family and she taught me to live and work with love forever. I pretty enjoy my family life with them.

Last but not least, I would like to thank the University of Technology Sydney and Beijing University of Posts and Telecommunications for co-funding this project. Without their support, I could not carry out my research and finish it without worry.

I also want to thank myself for my persistence and hard work. I have been strong to keep up my research going on without being disturbed by the outside world including Covid. Luckily, I held on rather than giving up.

Qianwen Ye

Sydney, Australia, 2023

# List of Publications

The following publications are the basis of this thesis [1–4]:

- Chapter 3

**Qianwen Ye**, Xiaochen Fan, Gengfa Fang, Hongxia Bie. Exploiting temporal dependency of RSS data with deep learning for IoT-oriented wireless indoor localization. Internet Technology Letters. 2022;e366.

- Chapter 4

**Qianwen Ye**, Xiaochen Fan, Hongxia Bie, Xudong Song, Rajan Shankaran, Capsloc: A Robust Indoor Localization System with Wi-Fi Fingerprinting using Capsule Networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1-6.

- Chapter 5

**Qianwen Ye**, Hongxia Bie, KuanChing Li, Xiaochen Fan, Liangyi Gong, Xiangjian He, Gengfa Fang, EdgeLoc: A Robust and Realtime Localization System towards Heterogeneous IoT Devices, IEEE Internet of Things Journal 9 (5) (2022) 3865-3876.

- Chapter 6

**Qianwen Ye**, Xiaochen Fan, Hongxia Bie, Deepak Puthal, Tao Wu, Xudong Song, Gengfa Fang, SE-Loc: Security-Enhanced Indoor Localization with Semi-Supervised Deep Learning, in IEEE Transactions on Network Science and Engineering, doi: 10.1109/TNSE.2022.3174674.

Other publications during the Ph.D. candidature [5–7]:

- Xudong Song, Xiaochen Fan, Xiangjian He, Chaocan Xiang, **Qianwen Ye**, Xiang Huang, Gengfa Fang, Liming Luke Chen, Jing Qin, Zumin Wang, CNNLoc:

Deep-Learning Based Indoor Localization with Wi-Fi Fingerprinting, in 'Proceedings of IEEE International Conference on Ubiquitous Intelligence and Computing (UIC)' (2019): 589-595.

- Xudong Song, Xiaochen Fan, Chaocan Xiang, **Qianwen Ye**, Leyu Liu, Zumin Wang, Xiangjian He, Ning Yang, Gengfa Fang, A Novel Convolutional Neural Network Based Indoor Localization Framework With Wi-Fi Fingerprinting, in 'IEEE Access' 7 (2019): 110698-110709.

- Shuang Lai, Xiaochen Fan, **Qianwen Ye**, Zhiyuan Tan, Yuanfang Zhang, Xiangjian He, Priyadarsi Nanda, FairEdge: A Fairness-Oriented Task Offloading Scheme for IoT Applications in Mobile Cloudlet Networks, in 'IEEE Access' 8 (2020): 13516-13526

# Contents

# List of Table

# List of Table

# Abbreviations

| | |
|---|---|
| Adam | Adaptive Moment Estimation |
| AE | AutoEncoder |
| AOA | Angle of Arrival |
| AP | Access Point |
| BLE | Bluetooth Low Energy |
| CapsNet | Capsule Network |
| CDF | Cumulative Distribution Function |
| CNN | Convolutional Neural Network |
| Conv | Convolutional |
| CS | Compressive Sensing |
| CSI | Channel-State Information |
| CSS | Chirp Spread Spectrum |
| DAE | Denoising AutoEncoder |
| dBm | Decibel-milliwatts |
| DBN | Deep Belief Network |
| DDoS | Distributed Denial of Service |
| DNN | Deep Neural Network |
| DR | Dead-Reckoning |
| FC | Feature-Capsule |

| | |
|---|---|
| FM | Frequency Modulation |
| GANs | Generative Adversarial Networks |
| GPS | Global Positioning System |
| GSM | Global System for Mobile Communications |
| ILBSs | Indoor Location-based Services |
| IoT | Internet of Things |
| IP | Internet Protocol |
| IQR | Interquartile Range |
| km | Kilometer |
| KNN | K-Nearest Neighbor |
| LBS | Localization-based Services |
| LEDs | Light Emitting Diodes |
| LoRa | Long Range Radio |
| LOS | Line of Sight |
| LPWAN | Low Power Wide Area Network |
| MAC | Medium Access Control |
| MitM | Man-in-the-Middle |
| MLPs | Multilayer Perceptrons |
| ms | millisecond |
| mW | MilliWatts |
| NLOS | Non-Line-of-Sight |
| NN | Neural Network |
| Normpdf | Normal Distribution Probability Density Function |

| | |
|---|---|
| PC | Primary-Capsule |
| PCC | Pearson Correlation Coefficient |
| ReLU | Rectified Linear Unit |
| RF | Radio Frequency |
| RFID | Radio Frequency Identification |
| RNN | Recurrent Neural Network |
| RPs | Reference Points |
| RSS | Received Signal Strength |
| RTT | Round Trip Time |
| SAE | Stacked Autoencoder |
| SSD | Signal Strength Difference |
| SSID | Service Set Identifier |
| SVM | Support Vector Machine |
| UWB | Ultra-Wideband |
| VAE | Variational Autoencoder |
| Wi-Fi | Wireless Fidelity |

# Abstract

In recent years, localization-based Internet of Things (IoT) applications have been developed and deployed, such as interactive and personalized routing, car localization in underground parking systems and patient emergency localization. However, in indoor environment, Global Positioning System signal is not available because it is very sensitive to occlusion. Many researchers have been focusing on utilizing other technologies such as Wi-Fi (Wireless Fidelity), Radio Frequency Identification, Bluetooth and so on for localization services. Among these technologies, Wi-Fi has been most widely utilized for indoor localization due to its low cost and wide availability. There are various signal measurements for the Wi-Fi-based indoor localization such as Received Signal Strength (RSS), Time of Arrival, Time Difference of Arrival, Round Trip Time, Angle of Arrival, and Channel-State Information. However, RSS remains the most popular signal measurement used in Wi-Fi-based localization solution compared to other measurements, especially for localization of low-cost IoT devices with limited computing and storage resources.

However, RSS-based indoor localization possesses many challenges due to multipath effects and noise, environment dynamics, device heterogeneity, limited high-quality data, and security. To overcome these challenges, in this thesis, RSS fingerprinting-based indoor localization methods are developed using machine learning methods and deep learning methods. For RSS time-series data, the system of Kalman-DNN

exploits the temporal dependency of these data by integrating the Kalman filter with deep neural networks, and experiment results validate effectiveness of the Kalman-DNN system. However, for single RSS readings vector, a system called CapsLoc is proposed, which is an RSS fingerprinting-based indoor localization system based on CapsNet (Capsule Network). The experimental results show that CapsLoc can achieve accurate indoor localization, which outperforms some traditional machine learning methods and existing deep learning methods. Especially for heterogeneous IoT devices, RSS can be affected by superimposed challenges, i.e., device heterogeneity, database problem and energy efficiency. In order to improve localization speed, EdgeLoc is proposed based on CapsNet and edge computing technology. Experiment results show that EdgeLoc outperforms state-of-the-art deep learning methods in performance of the localization accuracy and average positioning speed.

Considering security issues in localization where malicious attacks at APs (Access Points) exist, a solution of *SE-Loc* is proposed for RSS fingerprinting-based indoor localization utilizing the deep learning methods. Extensive experiments show that *SE-Loc* demonstrates superior performance on secure indoor localization over the baseline methods. To address challenges including the multipath effects and noise, the environment dynamics, the device heterogeneity, data limitation, database problem and even malicious AP attacks, deep learning-based indoor localization methods are proposed. In the future, it is necessary to develop security-enhanced deep learning techniques when facing other various security problems such as AP hijacking, jamming, and man-in-the-middle attack.

# Chapter 1

# Introduction

Based on the rapid development of Internet of things (IoT), localization-based service (LBS) becomes key technology component of many indoor IoT applications. For example, users can be navigated by IoT devices with LBS inside a building with high accuracy and low delay. Even when confronted by the security threats like malicious access point (AP) attacks, IoT devices can still support robust and secure indoor localization for users. Currently, there are several open challenges including high accuracy, low computation complexity, and security in indoor localization. This chapter first presents the research background. Then, this thesis deals with the motivation and challenges in indoor localization and deep learning, separately. At last, the structure of this thesis is summarized.

## 1.1 Research Background

Nowadays, many IoT-based applications require indoor localization as shown in Fig 1.1. There are a wide range of applications for localization of device or user such as navigation in shopping mall, medicine localization in hospital, smart device localization in smart house or building, car localization in underground parking and emergency services in fire house and indoor sports place [8, 9].

The relationship between indoor localization signals/technologies, measurements

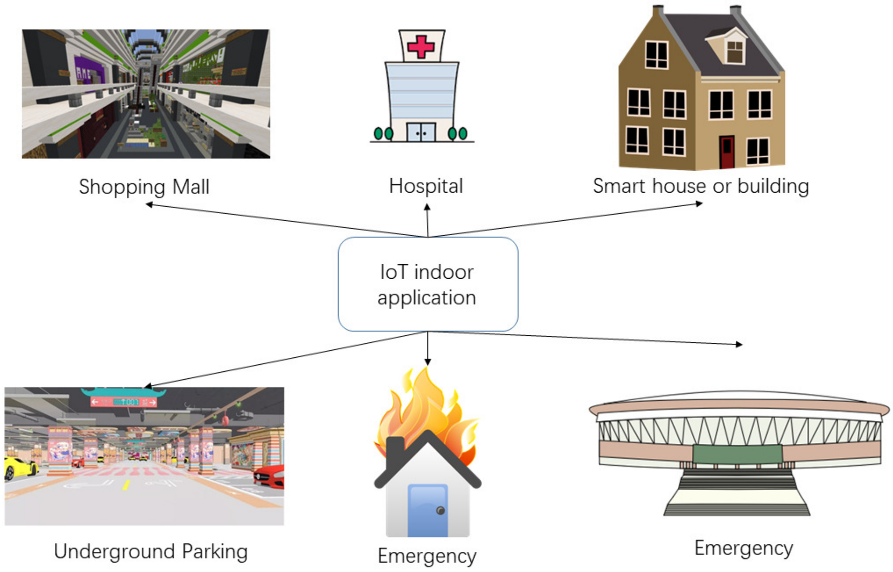


**Figure 1.1 .** IoT application scenarios requires indoor localization.

and methods is illustrated in Fig. 1.2. There are many indoor localization technologies of IoT applications, which are mostly based on wireless communication technologies including Wi-Fi (Wireless Fidelity), Radio Frequency Identification (RFID), Bluetooth, Ultra-Wideband (UWB), Ultrasound, Frequency Modulation (FM) Radio, Zigbee, Long Range Radio (LoRa), Mobile Networks (GSM (Global System for Mobile Communications), 3G, 4G) [10]. Among these technologies, Wi-Fi is popular since it requires no additional infrastructure, and is cost-effective to be accessed everywhere in indoor environment. Wi-Fi systems can carry out a variety of localization-signal measurements for localization purpose, including Received Signal Strength (RSS), Time of Arrival (TOA), Time Difference of Arrival (TDOA), Round Trip Time (RTT), Angle of Arrival (AOA), and Channel-State Information (CSI). The most ubiquitous Wi-Fi measurement is RSS, because it is hard to implement

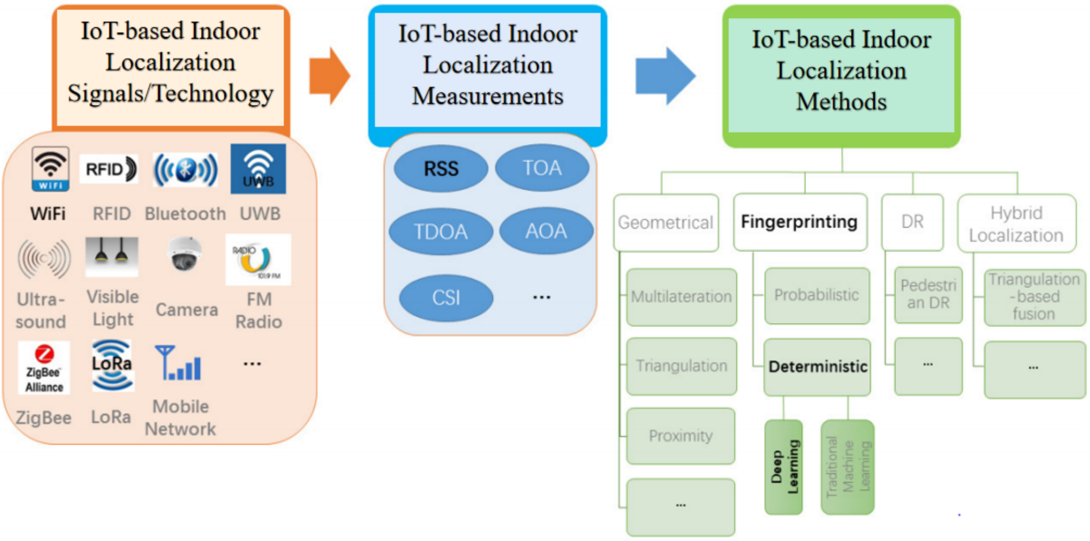


**Figure 1.2 .** The relationship between indoor localization signals, measurements, methods.

some functions when other measurements are utilized.

Many methods have been proposed for indoor localization including geometrical localization methods, fingerprinting-based methods, dead-reckoning (DR) and hybrid localization methods [8, 11]. However, DR methods, such as pedestrian DR, need additional inertial sensors for relative location estimation. Hybrid localization methods integrate multiple types of measurements or signals. For example, triangulation-based fusion enhances localization accuracy via combining several kinds of measurements including RSS, TOA, TDOA, and AOA, which requires two or more kinds of hardware components with extra cost. Since RSS can be easily accessed, it has become the most popular signal measurement for geometrical localization and fingerprinting-based methods. Geometrical localization methods include multilateration, triangulation and proximity *etc.* In some areas (*i.e.*, indoor and outdoor open spaces), it is more suitable to use these methods with the ability to model and parameterize explicitly. For example, many researchers utilize multilateration and triangulation for IoT node location estimation, which is based on locations of at least three APs and their distances to the IoT node. Its basic principle is to estimate the circles for two-dimensional indoor localization [8], which is based on estimated distances to APs. The path loss models are usually utilized to estimate the distance from an AP to a node in complex indoor scenarios. However, environmental factors, such as multipath loss, environment dynamics [12], measurement offset from device heterogeneity [13] and even third-party attacks [14] are sources of error, which are to be modeled and are hard to be mitigated.

In contrast, RSS fingerprinting-based methods can be utilized in complex indoor scenarios that are hard to be parameterized. RSS fingerprinting-based methods are made up of two parts: offline training and online localization. In the offline training phase, RSS training data are stored and utilized to learn the localization model between anchor locations and RSS fingerprinting data collected at an anchor from the APs. In the online localization phase, RSS data are fed into the trained matching model from the offline phase to estimate the node locations [15]. There are two types of RSS fingerprinting-based methods including deterministic and probabilistic. As to deterministic ones, they divide the interior space into units to create a radio map and obtain the estimated location by looking for the best match between the RSS data and the RSS radio map. As to probabilistic methods, namely distributed-based ones, the radio map is constructed utilizing the RSS distribution based on AP, and then the location of the node is estimated utilizing the probability distribution function.

Deterministic localization algorithms, such as K-Nearest Neighbor (KNN) [16–18] (relatively simple and popular), and Support Vector Machine (SVM) [19] are very popular. However probabilistic localization algorithms, such as Naïve Bayes and BayesNet [17] are more complex and rely on likelihood functions [18]. Nowadays, as the development of deep learning techniques, many researchers are focusing on deterministic localization utilizing deep learning techniques rather than traditional machine learning techniques such as KNN [20]. Conventional machine learning methods are inadequate to deal with raw RSS data and transform the initial data into a precise feature or representation vector for location classification.

In indoor localization, deep learning methods are available for kinds of tasks including dimension reduction of radio maps, feature extraction, regression, classification, and forecasting of users' locations under various and complex environments, even facing security challenges [9]. Generally, main deep learning methods are the Neural Network (NN)-based ones. For this reason, these models are often referred to deep neural networks (DNN), which are composed of layers of interconnected nodes [21], which are made up of an input layer, one or more hidden layer(s), and an output layer. Typically, there are three kinds of deep learning: supervised, unsupervised, and reinforcement learning. In supervised learning, all training data are labeled. In unsupervised learning, all training data are unlabeled. In reinforcement learning, to achieve a goal, an agent must learn to run or perform actions in an unknown and certain complex space. And it gets a reward in return.

As to another type of deep learning, *i.e.*, semisupervised learning, among all data, the training ones are labeled and the rest ones are unlabeled. Among deep learning methods, the most popular one is supervised learning. During a training process, it is necessary to compare the practical output with the predicted one out of the network. Then an error between these two values is calculated by an objective function. The adjustable parameters, namely weights, are updated to reduce the error. A classical supervised learning model might consist of hundreds of millions of those adjustable weights and labeled data for training a network. In the last decade, supervised methods such as DNN [22, 37, 38] and convolutional neural network (CNN) [6, 40] have been widely utilized for indoor localization. In addition, in some situations, semisupervised learning methods are needed including Stacked Autoencoder (SAE) [6], and Denoising AutoEncoder (DAE) [22]. For example, when suffering from attacks, it is a challenge to set up a secured learning system only with the labeled data. Because the unlabeled data are usually abundant and may be changed by the attackers, Therefore, it is hoped that learning performance can be enhanced with the appropriate amount of unlabeled data and labeled data.

## 1.2 Motivation and Challenges

This thesis discusses some of the significant challenges and motivations of Wi-Fi RSS fingerprinting-based indoor localization methods utilizing deep learning technology:

- **Multipath Effects and Noise.** The existence of multipath effects is one of essential challenges of indoor localization. Based on inherent nature of the Wi-Fi signals, Wi-Fi radio signal can be reflected, refracted and diffracted by the buildings, objects, and sometimes even human beings. RSS data can be utilized to calculate the distance between APs and mobile devices, while collecting these data is hard in existence of multipath effects. It is an essential influencing factor of indoor localization accuracy. The RSS data needs to be processed for accurate location estimation with methods that can minimize or eliminate negative effects of multipath.

- **Environment Dynamic.** The characteristics of an indoor environment are important factors influencing indoor localization. The performance of RSS-based methods is related to the dynamic changes of the environment in the interior space including the number of walls and ceilings, the different placement of furniture as obstacles, and the number of people at different times. It is necessary to take all these factors into account when utilizing suitable indoor localization methods. In other words, RSS is time and location dependent according to environment settings. Therefore RSS-based deep learning methods are valuable with resistance to environment dynamics.

- **Data Limitation.** A single RSS reading vector is one-dimensional RSS data collected from multiple APs. Deep learning models do not really reveal the relationships among the features of raw RSS data, which is a limitation of the data. Typically, deep learning models are preceded by data pre-processing algorithm, which processes raw RSS data and even ameliorates data limitation.

- **Database-related Problem.** For some of the publicly available databases to be discussed in the next section, there are a number of drawbacks including lack of validation, obsolescence, and limited data samples. Due to the above, a private radio map can be built and utilized with drawbacks including low coverage space, limited application scope, and low number of APs. For example, AP deployment map in

UJIIndoorLoc database and Beijing University of Posts and Telecommunications (BUPT) database are totally different, with 520 APs for UJIIndoorLoc database and 6 APs for BUPT implementation. In order to address the above situation, AP selection method is necessary when the number of deployed APs is very large.

- **Device Heterogeneity.** RSS measurements collected on different devices transmitted from APs are different. This is a main challenge for a universal indoor localization method. Most manufacturers have different implementations of hardware components and computation abilities. The device heterogeneity creates a new challenge in the adoption of an indoor localization method.

- **Energy Efficient.** Deep learning model has kinds of indispensable performance including training time and positioning time. During model training process, model is trained using labeled RSS data. During localization stage, a trained model estimates the location given the RSS data as input. For deep learning-based localization systems, achieving high accuracy at a low-energy consumption is difficult in a battery. In order to achieve high localization accuracy, training and execution of the deep learning models need big computation consumption. In order to address this problem, edge computing theory is applied when designing a localization system, where training processing can be put on the edge server and only localization execution based on the trained deep learning model runs on IoT devices.

- **Security.** Most users are reluctant to share data related to their location. The reason is that user and device location is pretty sensitive information as part of security issue of any IoT applications. Even a malicious node can infiltrate a system and perform system attacks in indoor localization that can certainly affect overall performance of indoor localization-based applications. A secured indoor localization method should be robust to attacks. Thus, it is necessary to design secured indoor localization system under malicious attacks.

Chapter 3 to Chapter 6 of this thesis has attempted to address these issues by designing indoor localization solutions utilizing Wi-Fi RSS fingerprinting-based deep learning methods for IoT devices and applications.

## 1.3 Thesis Overview

Based on the challenges this thesis discussed above, technical part of this thesis consists of four chapters as shown in Fig. 1.3. In Chapter 3, in order to tackle the challenges of multipath effects, noise and environment dynamic, a novel indoor localization method is presented, which exploits temporal dependency of RSS time-series data by utilizing Kalman filter with DNNs. In Chapter 4, to address typical challenges with data limitation, CapsLoc is proposed, which is based on Capsule Network (CapsNet) and RSS fingerprinting technology. For situations with database problem together with device heterogeneity and energy efficiency issues, EdgeLoc is proposed in Chapter 5, which is based on the CapsNet, RSS fingerprinting, and edge computing. A deep learning-based *SE-Loc* solution is proposed in Chapter 6, considering security challenge as discussed above.

## 1.4 Thesis Outline

The structure of this thesis is described as follows:

**Chapter 2**

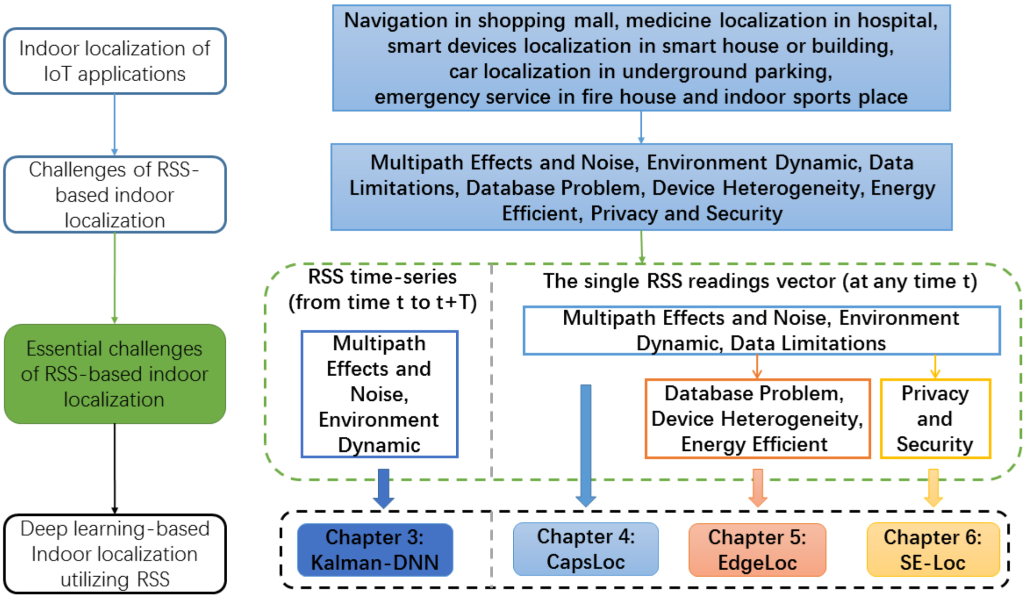

Indoor localization of IoT applications
Challenges of RSS-based indoor localization
Essential challenges of RSS-based indoor localization
Deep learning-based Indoor localization utilizing RSS
Navigation in shopping mall, medicine localization in hospital, smart devices localization in smart house or building, car localization in underground parking, emergency service in fire house and indoor sports place
Multipath Effects and Noise, Environment Dynamic, Data Limitations, Database Problem, Device Heterogeneity, Energy Efficient, Privacy and Security
RSS time-series (from time t to t+T)
Multipath Effects and Noise, Environment Dynamic
The single RSS readings vector (at any time t)
Multipath Effects and Noise, Environment Dynamic, Data Limitations
Database Problem, Device Heterogeneity, Energy Efficient
Privacy and Security
Chapter 3: Kalman-DNN
Chapter 4: CapsLoc
Chapter 5: EdgeLoc
Chapter 6: SE-Loc

**Figure 1.3 .** An overview of the thesis structure.

The state-of-the-art indoor localization technologies, measurements, and methods are reviewed and discussed in this chapter, followed by literature review of deep learning methods.

**Chapter 3**

RSS data are not only prone to multi-path reflections but also sensitive to time-varying environmental dynamics, which is one of the basic challenges of indoor localization. For example, RSS fluctuated sharply as people moving out of the room for lunch. In contrast to existing solutions focusing on spatial features of RSS, this chapter presents an innovative indoor localization method by exploiting the temporal dependency of RSS time-series data and integrating Kalman filter with DNNs. Experiment results show that with the same mean localization time, the proposed Kalman-DNN model outperforms the Kalman-CNN model on localization accuracy.

**Chapter 4**

To achieve high localization accuracy with Wi-Fi fingerprinting, CapsLoc, a robust indoor localization system based on capsule networks is proposed. Specifically, the proposed capsule network model can efficiently extract hierarchical structures from Wi-Fi RSS fingerprint, which fluctuates due to multipath effects, noise and environment dynamics. This chapter then conducted experimental field test with over 33600 data readings. experimental results show that CapsLoc outperforms conventional machine learning methods (KNN and SVM) and existing deep learning methods (CNN and SAE-CNN).

**Chapter 5**

This chapter proposes EdgeLoc, a robust and real-time indoor localization system considering heterogeneous IoT devices to solve significant challenges, such as RSS variances caused by hardware heterogeneity, database problem, and energy efficiency. Extensive field experiments are conducted to validate the effectiveness of EdgeLoc with a large-scale Wi-Fi fingerprint dataset. The results show that EdgeLoc outperforms the state-of-the-art SAE-CNN method in localization accuracy.

**Chapter 6**

This chapter proposes *SE-Loc*, a deep learning-based localization method to enhance resiliency and security of wireless indoor localization and improve the reliability of localization. The solution of *SE-Loc* consists of two parts: (1) AP selection for processing initially contaminated APs, and (2) a deep learning model based on a denoising autoencoder and convolution neural networks for feature learning and location matching. Extensive experiments show that *SE-Loc* demonstrates superior performance on secure indoor localization over the baseline methods. When confronting up to 100 malicious attacking APs in the UJIIndoorLoc database, *SE-Loc* can still achieve lowest average localization error with respect to other baselines.

**Chapter 7**

This chapter presents conclusions of this thesis and highlights potential research work for further investigations.

# Chapter 2

# Methods

## 2.1 Indoor Localization of IoT Applications

### 2.1.1 Indoor Localization Technologies

In indoor localization of IoT applications as shown in Fig. 1.1, many existing technologies are utilized to provide indoor localization. Radio communication technologies including Wi-Fi, Bluetooth, ZigBee, UWB, and RFID are presented first, followed by mobile network, and Ultrasound. Finally, several emerging technologies like camera and visible light are discussed. Among these technologies, Wi-Fi is the most widely utilized for indoor localization because of its wide deployment and availability[23].

1)RFID

RFID is utilized to transmit and store data from a transmitter to any radio frequency (RF) compatible circuit via electromagnetic transmission. The RFID systems can be divided into two types: active and passive. The active RFIDs in the Ultra High Frequency and microwave frequency range are not easy to implement on most portable user devices. Passive RFIDs have limited communication range (1-2m) without battery, which is unsuitable for indoor localization [24]. So RFID is not widely utilized for IoT indoor localization.

2)Bluetooth

Bluetooth (or IEEE 802.15.1) comprises the physical and Medium Access Control (MAC) layers specifications for connecting different fixed or portable wireless devices in a specific personal area. The most advanced version of Bluetooth is Bluetooth Low Energy (BLE), namely Bluetooth Smart. It is sponsored by Apple and is being distributed in the form of beacons [10]. BLE offers advanced data rate of 24 Mbps and the coverage range of 70-100 meters, which is energy efficient than older versions and narrower than the coverage range of Wi-Fi inside building. So BLE is less utilized for indoor localization as compared to Wi-Fi.

3)ZigBee

ZigBee is a specification implemented based on IEEE 802.15.4 standard, focusing on physical and MAC layer. In fact, it is also a low-rate wireless personal area network. It can realize the design requirements of applications with low power consumption but without large data throughput. ZigBee nodes can achieve signal coverage of up to 100m in open areas, but generally 20m to 30m in indoor environments [25]. The accuracy of ZigBee is very high, but unfortunately, for main user devices it is not available. Hence it does not work well for indoor localization.

4)UWB

UWB is a short-range radio technology that transmits short pulses with less than 1ns over a large bandwidth with more than 500 MHz with a very low duty cycle. Although UWB is less sensitive to multipath effects, it is subject to the Non-Line-of-Sight (NLOS) effect with cover range of 10-20 m [26]. Meantime, the development of UWB standards has been slow (although UWB has been initially proposed for use in personal area networks). This rate of development limits the marketing of UWB in portable user devices and consumer goods, especially as a standard. Therefore, UWB is not suitable for indoor localization.

5)Mobile Network

With appropriate program, Mobile networks including 3G, 4G, 5G, and GSM are able to realize a very precise localization with low-cost since there is no extra infrastructure and hardware equipment needed. However, this technology is mostly utilized for outdoor localization [26].

6)Ultrasound

The ultrasound signals are utilized for location estimation of the targeted node including a mobile device by the emitter tags from the wireless ultrasonic beacons. Main ultrasound tracking systems incorporate other technologies for distance estimation between the transmitter and receiver. The difference of arrival time is extracted by correlation processing of two received signals, and then makes distance estimation [27]. Usually, the necessary synchronization is achieved by combining the RF pulse with the ultrasonic signal transmission. However, the speed of sound varies significantly with humidity and temperature.

7)Camera

Received images of a scene from the camera are utilized for a presence detection, and to locate elements within the scene. The location is through a transformation between the scene image and angles of the camera as the contour tracking methods. As the emerging technology, cameras are widely utilized in localization of various precision levels, and its main application field is sub-millimeter. However, the coverage range of a localization camera is 1-10 m [25].

8)Visible Light

As a promising technology, visible light communication utilizes visible light of 400-800 THz for high-speed data communication and transmission, and its modulation and emission are mainly based on Light Emitting Diodes (LEDs). Usually, the receiver has to be equipped with a sensor to detect the LEDs' locations. There are always multiple LEDs in a room, which increases localization accuracy, but their emitted lights can not overlap with each other [10]. Meanwhile, it is high-cost to utilize this technology for localization.

9)Wi-Fi

The IEEE 802.11 standard, namely Wi-Fi, is primarily utilized to provide wireless Internet access of different devices in private, public and business scenarios. Initially, Wi-Fi has a reception range from about 200 meters (indoor) to about 5 kilometers (km) (outdoor) [26] in IEEE 802.11a. Wi-Fi-based methods are widely utilized for indoor localization due to the ubiquitous availability of Wi-Fi networks and Wi-Fi enabled devices.

### 2.1.2 Indoor Localization Measurements

Many localization methods are based on Wi-Fi signals for the user's location estimation. There are many measurements from Wi-Fi signals including RSS, TOA, TDOA, AOA, CSI, and RTT. Among these measurements, RSS is the most popular because it is the easiest parameter to measure.

1) RSS

RSS is one of the most accessible and popular measurement methods in Wi-Fi-based indoor localization. RSS refers to the signal power strength received from APs of Wi-Fi networks at the user devices, whose unit of measurement is milliWatts (mW) or decibel-milliwatts (dBm). However, RSS is an inaccurate measurement to estimate distance caused by shadowing, fading, reflections, scattering, and refraction, especially in complex indoor environment.

2) TOA

The principle of TOA-based method is that the distance between Wi-Fi AP and user device can be calculated according to the speed and propagation time of measured signal. Because the speed of Wi-Fi signal is the same as that of light (around 300000 km/s), namely it only takes about 20ns for Wi-Fi signals to transmit 6 meters [28]. In Fig. 2.1, TOA ($t_i$) from three different APs ($AP_i, i = 1, 2, 3$) is utilized to estimate distances ($d_i = vt_i$ with the signal velocity of $v$) between APs and a device. Multilateration is the traditional method of calculating the device location relative to the APs. However, it requires highly accurate synchronization between APs and a device. In synchronization, when there is a nanosecond error, the distance error is 30 cm. Furthermore, performance degradation caused by environmental impacts (*i.e.*, multipath and NLOS) becomes even more significant [8]. Because it is the most important and basic cause of indoor localization error.

3) TDOA

TDOA takes advantage of the difference in the propagation time of signals, which is measured on the device from different APs. TOA technology utilizes absolute travel time of the signal, which is different from TDOA. The TDOA from at least

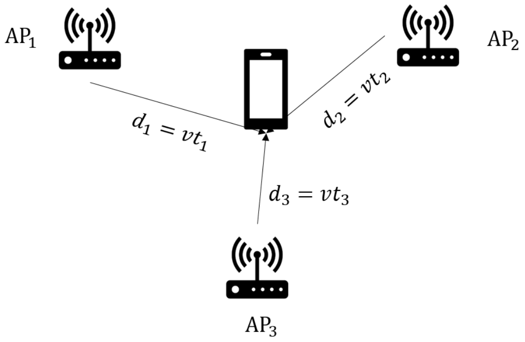


**Figure 2.1 .** TOA-based indoor localization.

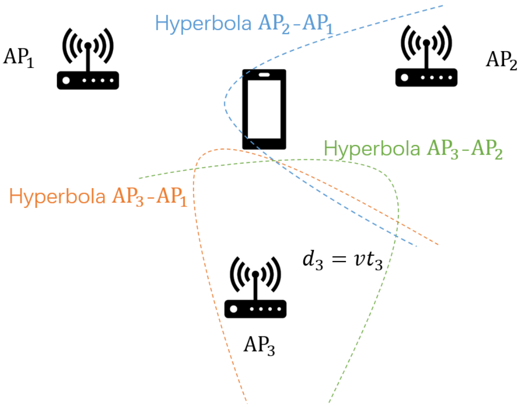


**Figure 2.2 .** TDOA-based indoor localization and proximity detection.

three APs is necessary for calculating the device location, which is the intersection of three (or more) hyperboloids [24]. The principle of location estimation utilizing three APs is shown in the Fig. 2.2. In this figure, the measurements collected from the APs are presented as hyperbolas for location estimation. TDOA can solve the synchronization error problem in certain degree, because it only requires synchronizing between APs. However, the NLOS propagation of Wi-Fi signals has a great impact on the performance of TDOA-based methods.

4) RTT/ two-way TOA/ Roundtrip TOA

RTT refers to round-trip signal propagation time, which is measured to calculate the distance between a device and APs. Ranging scheme for both TOA and RTOA is the same. After receiving the signal from an AP, the device responds to the AP, which then calculates total RTT. Compared to TOA, RTT doesn't require highly accurate synchronization between APs and the device. However, the accuracy of RTT estimation is affected by the homogeneous error sources of TOA [8]. Meantime, the response delay for a device also depends on its electronics and network protocol implementation, which cannot be ignored for indoor localization [24].

5) AOA

AOA is an indoor localization technology that utilizes receiving antenna array to estimate the signal propagation direction [28]. It is an enhanced ranging technology, which is based on measurements of distance and angle. Under ideal circumstances, it only requires two anchor nodes for location estimation [29]. While in contrast to RSS-based methods, it requires more accurate calibration and sophisticated hardware, and AOA is on the basis of LOS setting thus is often hard to measure required data because of multipath effects from indoor surroundings. In Fig. 2.3, AOA (with the same angle of $\gamma$) based on antenna array with the fixed spacing of $d$ can be utilized to estimate the user location.

6) CSI

CSI-based localization is also one of the enhanced ranging techniques. It is available to achieve precise estimation of received signal over the entire bandwidth of signal. Channel impulse response or its Fourier pair (namely, channel frequency response) is usually utilized by upper layers to calculate CSI. It is based on channel phase and amplitude response with various central frequencies and between different transmitting and receiving antennae pairs [30]. And it has higher granularity than that of RSS. However, it may not be available on off-the-shelf network interface controllers [8]. Meantime, the complexity of CSI technology is higher than RSS technology.

### 2.1.3 RSS-based Indoor Localization Methods

RSS-based indoor localization methods are divided into geometrical localization methods and fingerprinting-based methods.

For geometrical localization methods, RSS can be utilized to estimate distances between mobile devices and Wi-Fi APs. According to statistical analysis, RSS is negatively correlated with the distance, which is defined in path loss propagation model. Mostly, the distance $d$ between the mobile device and the Wi-Fi AP is calculated from log-normal shadowing model [31] as:

$$RSS_i[dB] = RSS_i(d_0) + 10\eta log(\frac{d_i}{d_0}) + n_{\sigma i}, \tag{2.1}$$

where $\eta$ is the path loss exponent, and $d_0$ is the reference distance. $RSS_i(d_0)$ is the reference path loss from the $i$th Wi-Fi AP, which is calculated using the path loss formula. The parameter $n_i$ is a zero-mean Gaussian random variable (in dB) with standard deviation $\sigma_i$. Then the estimated distance can be utilized to calculate the user location by traditional geometrical methods such as multilateration as shown in Fig. 2.4.

However, in some situations, it is inevitable to encounter security threats with malicious AP attacks at Wi-Fi APs deployed. In attack-resistant localization systems, an attack variable is introduced in RSS model as follows:

$$RSS_i[dB] = \begin{cases} I & \text{if there is the ith AP attack,} \\ RSS_i(d_0) + 10\eta log(\frac{d_i}{d_0}) + n_{\sigma i} & \text{otherwise,} \end{cases}, \tag{2.2}$$

$$I \sim U\{-100, 0\}$$

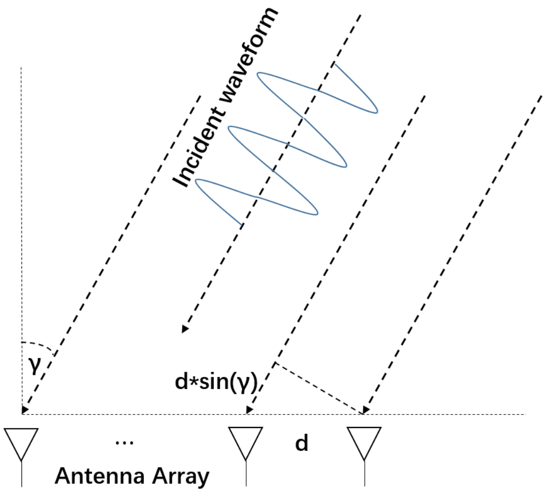


**Figure 2.3 .** AOA-based indoor localization.

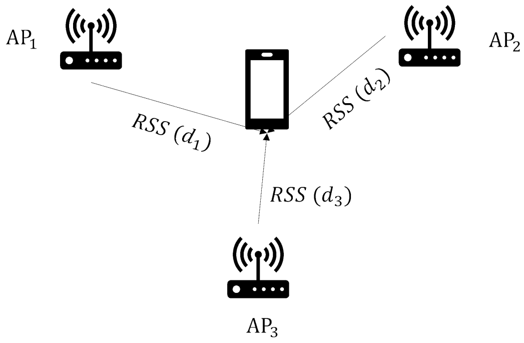


**Figure 2.4 .** RSS-based multilateration method.

where $I$ denotes attack perturbation from the $i$th Wi-Fi AP, and it is not zero only when the $i$th Wi-Fi AP attack is malicious [14]. Therefore, it is tough to realize the estimation of the geometrical distance in complex indoor environment, especially when facing security challenges. In these cases, RSS Fingerprinting-based methods as shown in Figure 2.5 are utilized rather than geometrical methods.

Two types of RSS fingerprinting-based methods are deterministic and probabilistic. In [17], deterministic algorithms such as KNN, sequential minimal optimization, probabilistic methods, and decision tree including BayesNet and Naïve Bayes, are machine learning algorithms. Among these algorithms, KNN is superior to all other methods in estimating location [17]. While in complex environments where data dimension is high and feature extraction is tough, deep learning technology has a broad application prospect in improving localization accuracy. In [20], a discriminant and adaptive NN is proposed, which updates the weights based on the

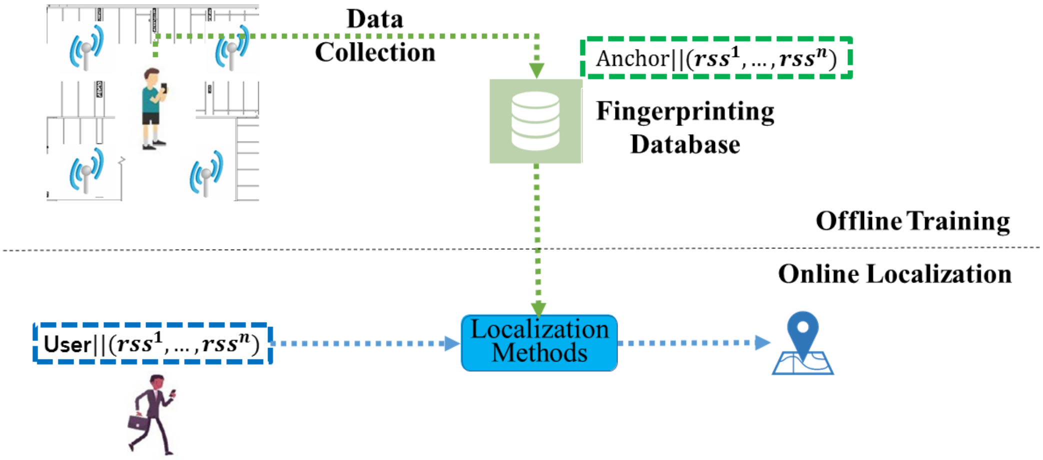


**Figure 2.5 .** The principle of Wi-Fi fingerprinting.

information extracted from the collected data (*i.e.* the discriminant component).

## 2.2 Deep Learning Technique

The word "deep" means that there are plenty of layers to be utilized to transform data. Or rather, deep learning is usually structured as multi-layer modules. The obedience principle of these modules is self-learning of non-linear input-output mappings. Each of them converts its input that increases both features' selectivity and invariance. With multiple non-linear layers, such as depths of 5 to 15, the network performs pretty complex functions on its inputs, which are sensitive to important details at the same time rather than to many separate variations such as the complex environment and dynamic human beings [32]. In indoor localization scenarios, many researchers utilize deep learning for multiple tasks, such as feature extraction, dimension reduction of radio maps, classification, regression, and forecasting of devices' locations [9]. The basic principle of a typical neural network is introduced, and deep learning-based methods are discussed.

### 2.2.1 Basic Principle of a Typical Neural Network

Fig. 2.6 shows a typical NN including one input layer (X), two hidden layers (H1 and H2) and one output layer (Y), where each layer constitutes a module by which gradients can be backpropagated. The equations are utilized to calculate the forward pass in NN. At each layer, the input of layer $z$ ($i.e.$ $z_k$) is calculated with a weighted sum of the outputs of next layer ($i.e.$ $y_i$). After that a non-linear function $f(.)$ is utilized to $z$ to calculate the output of this layer ($i.e.$ $y_k$). In brief, bias terms are omitted. The non-linear functions applied in NN contain the rectified linear unit (ReLU) $f(z) = max(0, z)$, which is widely utilized in these years, even the more traditional sigmoid, including the hyperbolic tangent, $f(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$ and logistic function, $f(z) = \frac{1}{1+\exp(-z)}$ [32]. Fig. 2.7 shows the back propagation process of a NN. At the output layer, error $E$ is calculated from the output of NN ($y_l$) and the true result ($y_{true}$). Then, the error derivative ($\frac{\partial E}{\partial y_l}$) corresponding to $y_l$ is computed by differentiating a cost function ($l_k$). At each hidden layer, the error derivative ($i.e.$ $\frac{\partial E}{\partial y_k}$) corresponding to the output of each layer ($i.e.$ $y_k$) is a weighted sum of the error derivatives ($i.e.$ $\frac{\partial y_k}{\partial z_k}$) corresponding to the total inputs ($i.e.$ $z_k$) calculated from the outputs of the above layer. The error derivative ($i.e.$ $\frac{\partial E}{\partial z_k}$) corresponding to the output is the error derivative ($i.e.$ $\frac{\partial E}{\partial y_k}$) multiplied by gradient ($i.e.$ $\frac{\partial y_k}{\partial z_k}$) of $f(z)$. For example, if a cost function of unit $i$ is $0.5(y_i - t_i)^2$ ($t_i$ is the

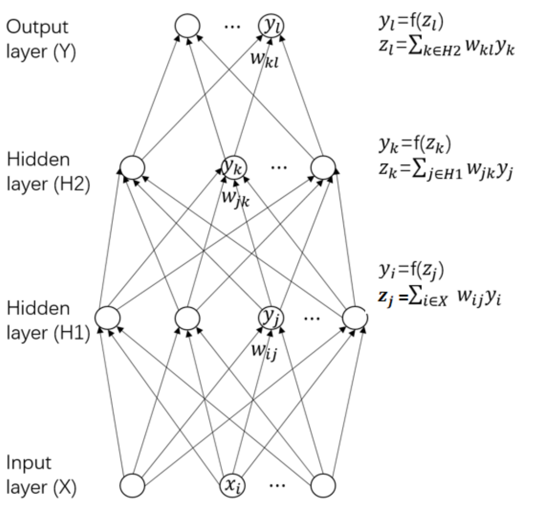


Figure 2.6 . Multilayer neural networks.

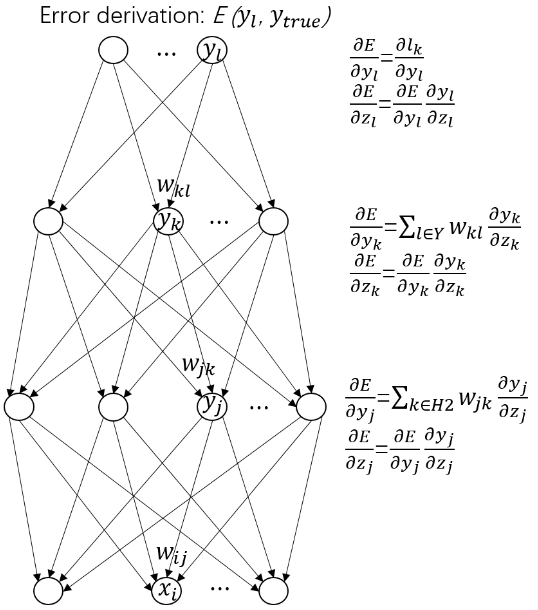

Error derivation: $E(y_l, y_{true})$
$\frac{\partial E}{\partial y_l}=\frac{\partial l_k}{\partial y_l}$
$\frac{\partial E}{\partial z_l}=\frac{\partial E}{\partial y_l}\frac{\partial y_l}{\partial z_l}$
$\frac{\partial E}{\partial y_k}=\sum_{l\in Y} w_{kl}\frac{\partial y_k}{\partial z_k}$
$\frac{\partial E}{\partial z_k}=\frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial z_k}$
$\frac{\partial E}{\partial y_j}=\sum_{k\in H2} w_{jk}\frac{\partial y_j}{\partial z_j}$
$\frac{\partial E}{\partial z_j}=\frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial z_j}$
$y_l$
$w_{kl}$
$y_k$
$w_{jk}$
$y_j$
$w_{ij}$
$x_i$

Figure 2.7 . Backpropagation.

target value), there will be $\frac{\partial E}{\partial y_i} = y_i - t_i$. Once the $\frac{\partial E}{\partial y_l}$ is given, the error-derivative of the weight $w_{jk}$ on the connection from unit $j$ in the layer below is exactly $y_j \frac{\partial E}{\partial y_l}$ [32].

### 2.2.2 Deep Learning-based Indoor Localization Utilizing Wi-Fi RSS Fingerprinting

Many researchers have utilized deep learning for Wi-Fi RSS Fingerprinting-based indoor localization in recent years. The key point is to find an exact match between the RSS data and custom grid points on a fingerprint map. However, RSS suffers from multipath effects, environment dynamics, device heterogeneity, database problem and even privacy and security. So, it is hard to achieve exact match by the traditional machine learning methods, like KNN and SVM due to limit learning ability. Many researchers have studied deep learning-based localization methods. Multilayer perceptrons (MLPs) [33, 34] and CNNs [33–35] are the baseline deep learning models, which are widely utilized for RSS fingerprinting-based indoor localization. In [34], CNN model outperforms MLP (typical DNN) on localization accuracy in indoor environment. In [33], a one-dimension CNN model (for feature extraction) combined with MLP model outperforms DNN model for location estimation. However, learning ability of CNN model is lower than that of CaspNet [36] because CapsNet can capture the hierarchical structure of entities in RSS data. Supervised deep learning methods such as DNN and CNN are utilized for database with large labeled data. But there are many databases with insufficient labeled data or randomly unlabeled data caused by the security concern. In this way, semi-supervised deep learning methods are more suitable. In particular, Generative Adversarial Networks (GANs) [37], AutoEncoder (AE) [35, 38–40], SAE [6], Variational Autoencoder (VAE) [41], and DAE [22] have been employed. These methods leverage augmented data for indoor localization. Then DNN [22, 37, 38] and CNN [6, 40] are utilized for location estimation. These deep learning-based methods have the ability of high computational power and feature extracting, but they are also energy-consuming with low accuracy. So light-weight and high-accuracy deep learning-based indoor localization methods are necessary.

## 2.3 Summary

In order to extract features from RSS data with fluctuations caused by factors such as multipath effects, noise and environment dynamic, a novel indoor localization method is presented in Chapter 3 [1], which exploits temporal dependency of RSS time-series data by integrating Kalman filter with DNNs. When considering extra challenge of data limitation, CapsLoc [2] is designed in Chapter 4 for RSS-based indoor localization based on CapsNet model. In applications where device heterogeneity and real-time requirement are considered together with other challenges, Edgeloc [3] is proposed in Chapter 5 for RSS-based indoor localization, which is based on the CapsNet model and edge computing. Security concern makes RSS data become more randomly available, and fluctuation of testing RSS data set can be different from that of training RSS data set. In order to solve the above issue, *SE-Loc* is proposed in Chapter 6 for security-enhanced indoor localization. Finally, the thesis is concluded in chapter 7 followed by possible future work.

# Chapter 3

# Kalman-DNN: Exploiting Temporal Dependency of RSS Data with Deep Learning for Resilient Indoor Localization

With ubiquitous demand for indoor location-based services (ILBSs) and pervasive deployment of Wi-Fi hotspots, wireless indoor localization has been widely studied by utilizing various Wi-Fi signal measurements. Most existing schemes leverage RSS of Wi-Fi signal to conduct cost-efficient indoor localization. However, RSS data are not only prone to multi-path effects, but also sensitive to time-varying environmental dynamics, making it quite daunting to achieve robust indoor localization. In contrast to existing solutions that focus on spatial features of RSS, this chapter exploits temporal dependency of RSS time-series data by integrating Kalman filter with deep neural networks. In particular, to tame time-varying noises and preserve valuable temporal features in RSS measurements, this chapter proposes a time-varying RSS filtering algorithm based on the Kalman filter and a refined post-processing module. Moreover, a deep learning model based on DNN is further utilized for effective feature extraction on one-dimension RSS fingerprints. The experiment results show that the proposed Kalman-DNN model improves at least 25% localization accuracy in comparison with conventional DNN model. Furthermore, with the localization time as 0.02 millisecond (ms), the Kalman-DNN model outperforms Kalman-CNN model in localization accuracy by at least 10%.

## 3.1 Introduction

With proliferation of portable IoT devices and penetration of wireless networks in public indoor space, indoor localization has become imperative to support a variety of ILBSs. To facilitate wireless indoor localization, researchers have explored many short-range communication technologies, including ZigBee, Bluetooth, Wi-Fi, cellular networks, as well as their combinations [42]. Due to its wide availability and ubiquitous accessibility, Wi-Fi has become one of the most attractive infrastructures for indoor localization. In particular, RSS of Wi-Fi has received intensive research interests from research community in achieving none-line-of-sight indoor localization. However, RSS-based indoor localization is inherently vulnerable to multi-path effects and environmental dynamics, which lead to signal reflections and even signal fading [43]. The above vulnerabilities may significantly compromise efficiency and accuracy of Wi-Fi RSS-based indoor localization.

Existing efforts mainly focus on taming external influence on RSS to enhance the performance of indoor localization. For instance, He *et al.* proposed a Gaussian regression model to compensate for frequency-dependent shadowing effects and multi-path in RSS [44]. To further reduce error, Katwe *et al.* presented an effective hybridization measurement of time of arrival and RSS [45]. While existing schemes can improve localization accuracy in some typical indoor scenarios, the fundamental limit of RSS's continuous dependency on environmental dynamics is still not fully addressed yet [46]. To overcome above limitations, RSS fingerprinting has been extensively applied to enable efficient data collection and radio map construction.

This chapter explores temporal dependency of RSS data through technologies of Kalman filter and deep learning. This chapter aims to achieve effective RSS signal processing and devise an RSS filtering algorithm based on Kalman filter theory. Then, a post-processing module is also proposed to compress RSS time-series data and reduce computation complexity, where it transforms original RSS input into dynamic-resistant RSS time-series fingerprints. This chapter exploits further employ a deep neural network (DNN) to extract useful representations from RSS time-series used in localization model training stage. This chapter conducts extensive experiments through real-world indoor localization testbed. Results show the benefits of combining Kalman filtering algorithm with deep learning to process RSS measurements for IoT-oriented indoor localization.

The rest of this section is organized as follows. Section 3.2 introduces system architecture of the proposed indoor localization system. Section 3.3 elaborates the algorithm of Kalman-DNN that exploits temporal dependency of RSS data. Section 3.4 presents experimental results. At last, Section 3.5 concludes this chapter.

## 3.2 Overview of Kalman-DNN System

As shown in Fig. 3.1, the system architecture of Kalman-DNN consists of 5 modules, *i.e.*, data collection module, RSS filtering module, data post-processing module, online testing module and localization module. Different from existing studies, this chapter exploits temporal dependency of fingerprints by utilizing RSS time-series data in feature extraction and model training processes. This chapter denotes each reference point (RP) as $L_i(i = 1, 2 \ldots I)$ and use $\{RSS^n_{T_k}, \ldots, RSS^n_{T_k+t}\}$

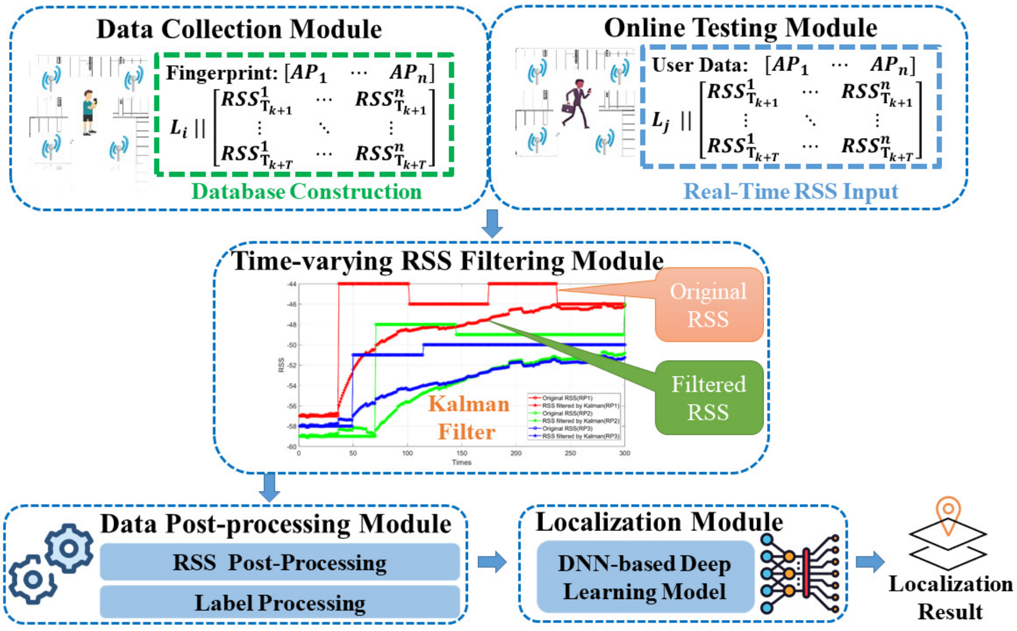

Data Collection Module
Fingerprint: $[AP_1 \cdots AP_n]$
$L_i \,||\, \begin{bmatrix} RSS^1_{T_{k+1}} & \cdots & RSS^n_{T_{k+1}} \\ \vdots & \ddots & \vdots \\ RSS^1_{T_{k+T}} & \cdots & RSS^n_{T_{k+T}} \end{bmatrix}$
Database Construction
Online Testing Module
User Data: $[AP_1 \cdots AP_n]$
$L_j \,||\, \begin{bmatrix} RSS^1_{T_{k+1}} & \cdots & RSS^n_{T_{k+1}} \\ \vdots & \ddots & \vdots \\ RSS^1_{T_{k+T}} & \cdots & RSS^n_{T_{k+T}} \end{bmatrix}$
Real-Time RSS Input
Time-varying RSS Filtering Module
Original RSS
Filtered RSS
Kalman Filter
RSS
Times
-44
-46
-48
-50
-52
-54
-56
-58
-60
0
50
100
150
200
250
300
Original RSS(RP1)
RSS filtered by Kalman(RP1)
Original RSS(RP2)
RSS filtered by Kalman(RP2)
Original RSS(RP3)
RSS filtered by Kalman(RP3)
Data Post-processing Module
RSS Post-Processing
Label Processing
Localization Module
DNN-based Deep Learning Model
Localization Result

**Figure 3.1 .** The system architecture of Kalman-DNN.

to represent raw RSS time-series measured from $n_{th}$ AP in time slot $T_k$ to $T_k + t$, where $k \in \mathbb{N}_+$.

During offline training procedure, raw RSS time-series are pre-processed by the time-varying RSS filtering algorithm to reduce noises caused by multi-path effects and environmental dynamics. Then, the filtered RSS time-series are further processed and compressed by the post-processing module to decrease computation complexity. After that, a DNN model learns temporal features from RSS time-series fingerprints with corresponding labeled locations. For online testing phase, this chapter utilizes raw RSS time-series collected by mobile users for localization. The main steps of RSS filtering and post-processing are the same as offline phase. Note that the trained DNN model with optimized parameters is employed to compute the similarity between the RSS time-series measurements and radio maps in

**Algorithm 1** The Time-varying RSS Filtering Algorithm

```
Input: The measured RSS value of the current state at time t: $\boldsymbol{y}(t)$; The estimated RSS value of the previous state at time t-1: $\widetilde{\boldsymbol{x}}(t-1)$; The smoothed estimated error covariance of the previous state at time t-1: $\widetilde{\boldsymbol{P}}(t-1)$; The system process variance matrix: $\boldsymbol{Q}$; The system noise covariance matrix: $\boldsymbol{R}$;
Output: The smoothed estimated RSS value of the current state at time t: $\boldsymbol{x}(t)$; The smoothed estimated error covariance at time t: $\widetilde{\boldsymbol{P}}(t)$;
Predict the prior estimate of RSS value at time t: $\widetilde{\boldsymbol{x}}(t|t-1)$ by Equation 3.3;
Predict the prior estimate of error covariance at the time t: $\widetilde{\boldsymbol{P}}(t|t-1)$ by Equation 3.4;
Predict the gain at time t: $K(t)$ by Equation 3.5;
Predict the smoothed estimated error covariance at time t: $\widetilde{\boldsymbol{P}}(t)$ by Equation 3.6;
Predict the smoothed estimated RSS value at time t: $\widetilde{\boldsymbol{x}}(t)$ by Equation 3.7;
return $\widetilde{\boldsymbol{x}}(t)$, $\widetilde{\boldsymbol{P}}(t)$;
```

the offline database. Finally, the Kalman-DNN outputs localization results with the best-match with RSS inputs.

## 3.3 RSS Filtering, Data Processing and Model Training

### 3.3.1 Time-varying RSS Filtering Algorithm

As shown in Fig. 3.2, for time $T_1$ and $T_2$, this chapter samples $T$ consecutive RSS time-series data, respectively. Notation $T$ means consecutive $T$ sampling points as

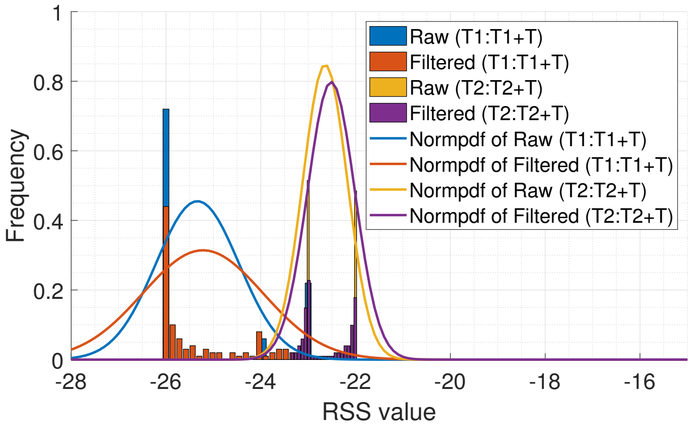


**Figure 3.2 .** The RSS time-series and the corresponding Normally distributed probability density function (Normpdf) before/after Kalman filtering.

shown in Fig. 3.3, where the solid line represents the raw data, and the dashed line represents the filtered data. It can be observed from RSS time-series data that there are different types of fluctuations in RSS values at $T_1$ and $T_2$. Caused by environmental dynamics, such random and abrupt changes in RSS measurements can significantly reduce the accuracy of indoor localization. This chapter aims to tame such fluctuations by proposing a time-varying RSS filtering algorithm based on Kalman filter. The Kalman filter-based algorithm takes measured RSS time-series of the current state (*i.e.*, a period of $T$) as an input, which is denoted by $\boldsymbol{y}(t) = (RSS_t^1, RSS_t^2, \dots RSS_t^n)$ and $(t = T_k + 1, T_k + 2 \dots T_k + T)$. Then, for RSS time-series data in the previous state, this chapter denotes its error covariance matrix as $\widetilde{\boldsymbol{P}}(t-1)$ and utilize it to predict the smoothed values of estimated RSS time-series of current state $\widetilde{\boldsymbol{x}}(t) = (\widetilde{RSS}_t^1, \widetilde{RSS}_t^2, \dots \widetilde{RSS}_t^n)$.

To this end, the state space model for the proposed Kalman filter can be written

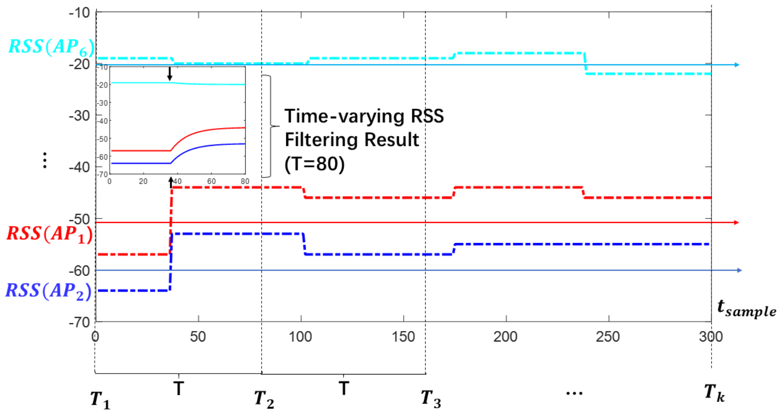


**Figure 3.3 .** At one location, the RSS time-series collected from several APs at 300 sample time points before/after Kalman filtering.

as

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + w(t-1), \tag{3.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) + v(t), \tag{3.2}$$

where $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ are state and measurement variances, respectively. $w(t)$ and $v(t)$ are the system noise and observation noise with covariance matrices $\boldsymbol{Q}_n$ and $\boldsymbol{R}_n$, respectively.

This chapter further combines system noise and observation noise to calculate the estimated RSS time-series of the current state with the following equations:

$$\widetilde{\boldsymbol{x}}(t|t-1) = \widetilde{\boldsymbol{x}}(t-1), \tag{3.3}$$

$$\widetilde{\boldsymbol{P}}(t|t-1) = \widetilde{\boldsymbol{P}}(t-1) + \boldsymbol{Q}, \tag{3.4}$$

$$K(t) = \widetilde{\boldsymbol{P}}(t|t-1)(\widetilde{\boldsymbol{P}}(t|t-1) + \boldsymbol{R})^{-1}, \tag{3.5}$$

$$\widetilde{\boldsymbol{P}}(t) = (1 - \boldsymbol{K}(t))\widetilde{\boldsymbol{P}}(t|t-1), \tag{3.6}$$

$$\widetilde{\boldsymbol{x}}(t) = \widetilde{\boldsymbol{x}}(t|t-1) + K(t)(\boldsymbol{y}(t) - \widetilde{\boldsymbol{x}}(t|t-1)), \tag{3.7}$$

where $\widetilde{\boldsymbol{x}}(t|t-1)$ and $\widetilde{\boldsymbol{P}}(t|t-1)$ represent posteriori state estimate and the error covariance matrix at time $t$, given measurements until time $t-1$. $\widetilde{\boldsymbol{x}}(t-1)$ and $\widetilde{\boldsymbol{P}}(t-1)$ represent posteriori state estimate and error covariance matrix at time $t-1$, given measurements until time $t-1$. $\boldsymbol{K}(t)$ is the Kalman gain, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are covariances of process and measurement noise, respectively.

Based on the above mathematical equations of a basic linear Kalman filter, this chapter further devises the time-varying RSS filtering algorithm and its pseudo-code in Algorithm 1. This chapter presets the initial error covariance matrix $\boldsymbol{P}(0)$ as [1], the noise covariance matrix $\boldsymbol{Q}$ as [0.001] and the observed noise covariance matrix $\boldsymbol{R}$ as [0.1].

### 3.3.2 Data Post-processing

***RSS Data Post-processing***

This chapter further proposes the post-processing module, which is designed to compress RSS data and reduce its computation complexity. First, this chapter derives the mean value of each set of $T$ RSS time-series samples, which is the mean value of $T$ RSS reading values collected from the same AP at the same place. Then, this chapter normalizes the calculated RSS values by

$$r_i = \begin{cases} 0 & RSS_i \text{ is none,} \\ 0.1 * (RSS_i - \min) & \text{otherwise,} \end{cases} \tag{3.8}$$

where $r_i$ is the normalized RSS value from AP $i$, $RSS_i$ is the raw RSS value from AP $i$, and min is the smallest RSS value in all the averaged RSS measurements.

***Label Processing***

To determine the label of RSS fingerprints at each reference point, this chapter divides the localization area into a number of zones. Each zone is a grid area with size of 1.6 × 1.6 $m^2$, which covers 4 RPs to reduce the number classification and computing complexity. To generate the label for each grid, this chapter adopts One-Hot Encoding method [47] to map each grid onto a One-Hot vector. Consequently, each individual grid represents a categorical variable, and the indoor localization task essentially becomes a classification problem across all grids with ground-truth fingerprints.

### 3.3.3 DNN Model Training

As shown in Fig. 3.4, the Kalman-DNN model consists of a multi-layered DNN model with multiple hidden layers. The proposed DNN consists of three types of layers, including input layer, hidden layers, and output layer. Based on the output of the previous layer, a non-linear function of hidden layers is defined as follows.

$$\boldsymbol{h}^{l^{(i)}} = f(W^{l^{(i)}}\boldsymbol{h}^{l^{(i-1)}}b^{l^{(i)}}), \tag{3.9}$$

where $W^{l^{(i)}}$ is the matrix of weights, indicating all the synaptic connections between each neuron of layer $l^{(i-1)}$. Each h neuron of layer $l^{(i)}$, $b^{l^{(i)}}$ is the bias vector of layer $l^{(i)}$, $\boldsymbol{h}^{l^{(i-1)}}$ is the output of the previous layer $l^{(i-1)}$, and $f(\cdot)$ is the activation function that calculates the non-linear relationship between layers [32].

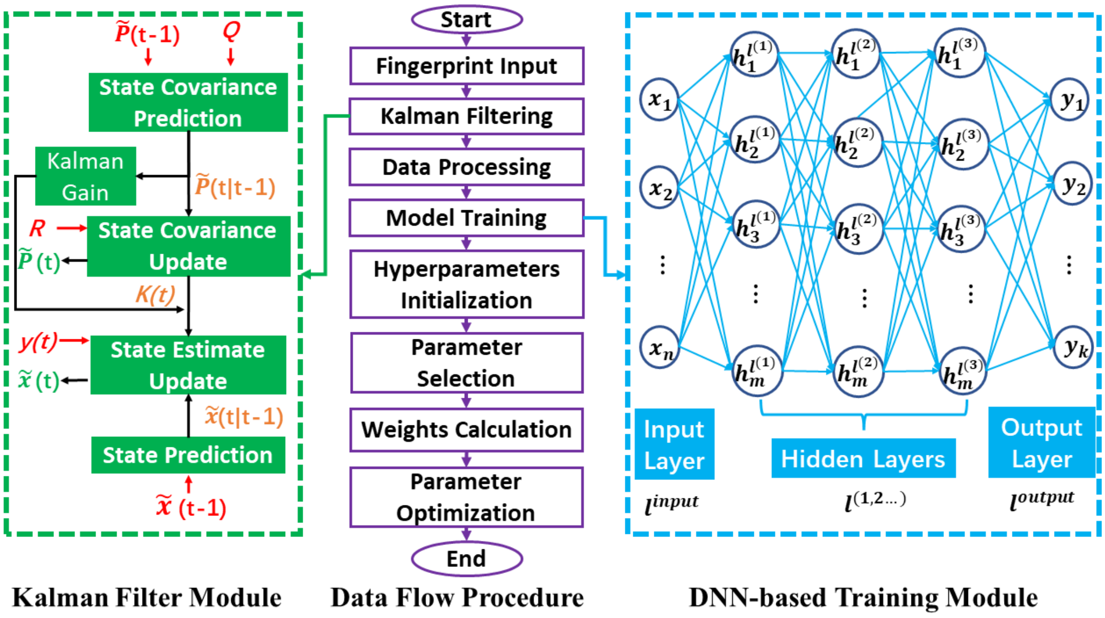


**Figure 3.4 .** The flowchart of the offline training phase and the architecture and parameters of the DNN model.

Fig. 3.4 presents the flowchart of the offline training phase for indoor localization and the parameters of the DNN model. For parameter tuning, this chapter conducts a grid search to find the best parameters to improve localization accuracy. Because it is easy to find the best matching of grid rather than RP's location, which means higher possibility to achieve accurate localization. This chapter also trains DNN model with different parameter settings for comparison purposes. This chapter chooses the rectified linear (*i.e.*, RELU) function as the activation function for the input and hidden layers. The output unit's activation function is the Softmax and its loss function is the categorical cross-entropy. The Softmax activation function is given a vector of raw outputs of the neural network and returns a vector of probability scores. The location classification is a multiclass one, then the output layer would have one node per class and a Softmax activation should be utilized. In the model training process, this chapter employs Adam (Adaptive Moment Estimation) as the optimizer of the proposed Kalman-DNN model.

## 3.4 Experimental Study

This chapter implements a real-world indoor localization testbed in the IoT lab of Beijing University of Posts and Telecommunications, as illustrated in Fig. 3.5. In this experimental environment, this chapter deploys 6 Wi-Fi APs to cover three lab rooms and a corridor area (totally 460 $m^2$). To achieve cost-efficient localization, this chapter places 2 TP-Link wireless APs in each room and set a number of reference points that are evenly distributed across each room. The distance between two adjacent RPs is $0.8m$ and this chapter measures RSS fingerprints at each RP for 300 times. The final dataset contains over $33,600$ fingerprinting samples, with 60% fingerprints as the training set and the rest 40% as testing set. The data are collected in the real world, but the localization system is simulated via Python. The proposed Kalman filter algorithm and DNN model are implemented in the Tensor Flow framework, using a Dell laptop with Intel Core i7-7600 CPU.

**Experiment evaluation indicators.** The experiment evaluation indicators of this chapter include localization error, average localization error, mean localization time. Localization error is the Euclidean distance between estimated coordinates and actual ones of the same location. Average localization error is the mean value of localization error of all testing data. Localization time is the time that one testing data is processed by all online modules. Mean localization time is the mean time of

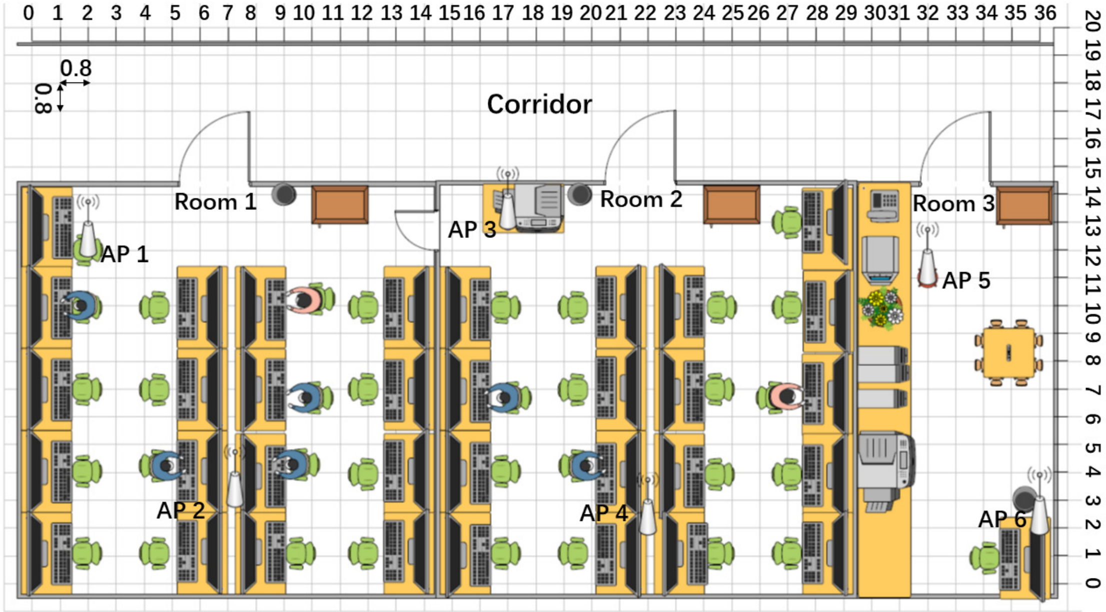


**Figure 3.5 .** The floor plan of the experiment field.

all testing data.

### 3.4.1 The Effect of DNN Parameters on Localization Errors

Fig. 3.6 shows the Cumulative Distribution Function (CDF) of localization errors () by Kalman-DNN model with different parameter settings. For instance, the model parameter 128-128-128-128 indicates the Kalman-DNN model that has 4 fully connected layers with 128 filters in each layer. Note that the final output layer of the Kalman-DNN is a fully connected neural network layer. From Fig. 3.6, It can be observed the localization accuracy is positively correlated to the number of filters in each layer. For instance, when the model parameter is 128-128-128-128, the localization errors for over 99% of testing data are under 2 $m$. Moreover, with the same number of filters, when the number of layers is reduced from 5 to 3, the localization accuracy significantly drops (*e.g.*, the localization errors are larger than 2 $m$ for over

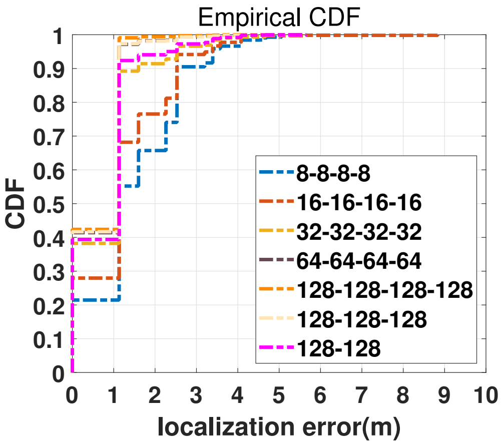


**Figure 3.6 .** The localization errors by Kalman-DNN with different parameter settings.

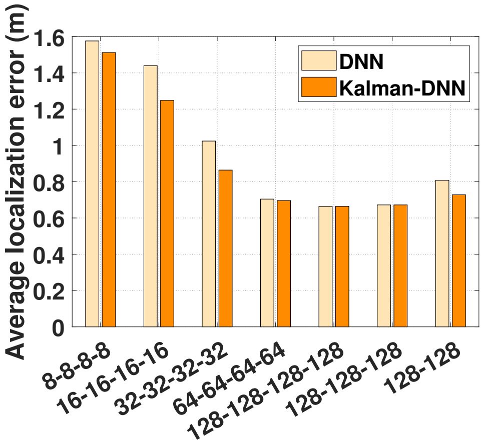


**Figure 3.7 .** Performance comparison of DNN and Kalman-DNN.

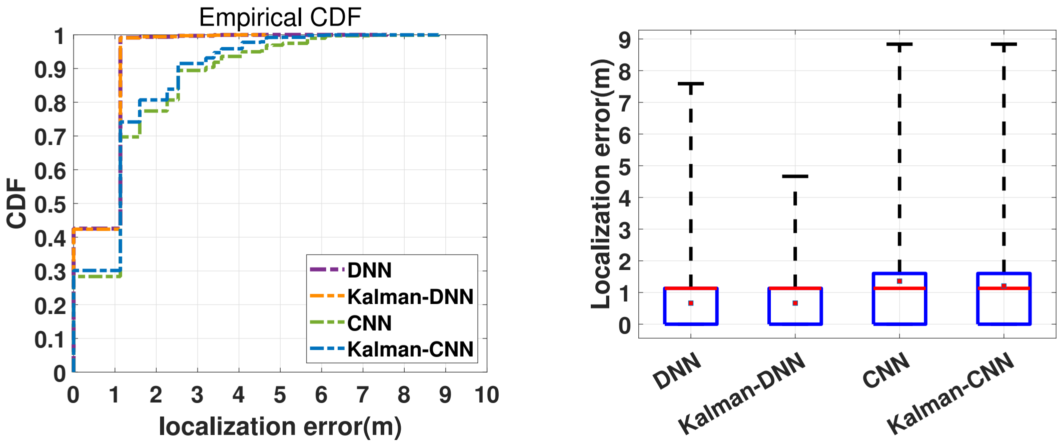
8% of the testing results).

### 3.4.2 Experiment Result of the RSS Filtering Algorithm

Recall from Fig. 3.2 that $T$ RSS time-series starting from $T1$ and $T2$ are different with noises, especially when confronting abrupt fluctuations caused by environmental dynamics. This chapter leverages a Kalman-based filtering algorithm to tame the above noises and fluctuations. The experimental results are shown in Fig. 3.7. This chapter shows average localization errors of conventional DNN model and the proposed Kalman-DNN model. In particular, when model parameters are set as 8-8-8-8, 16-16-16-16, 32-32-32-32, 64-64-64-64 and 128-128, the proposed Kalman-DNN model consistently outperforms the basic DNN model with reduction of 0.07 $m$, 0.2 $m$, 0.16 $m$, 0.01 $m$ and 0.08 $m$ in average localization errors, respectively. Although BUPT testbed has only 6 APs and is in small-scale localization scenarios, the proposed Kalman-DNN model has already made a remarkable difference in improving localization accuracy. This chapter also verified the significant improvement of Kalman-DNN through experiments with large-scale indoor localization datasets, such as UJIIndoorLoc dataset and Tampere dataset [6].

### 3.4.3 Performance Comparison of Different Localization Algorithms

To further evaluate the performance of Kalman-DNN that integrates temporal RSS features with deep neural networks, this chapter compares the localization performance of DNN, CNN, Kalman-DNN and Kalman-CNN in Fig. 3.8. In this experiment, this chapter sets the mean localization, *i.e.*, the mean computation time for an RSS fingerprint input in online testing, as 0.02 milliseconds. The CNN and Kalman-CNN models both have three layers, with 16 filters for convolutional operations. Similarly, the DNN and Kalman-DNN models both have three layers, with 128 filters in each layer. First, as shown in Fig. 3.8a, the Kalman-DNN achieves the best performance with localization errors smaller than 1.13 $m$ with 99% of the cases. By exploiting temporal dependency of RSS measurements, the Kalman-DNN model improves at least 25% and 10% in accuracy in comparison with the conventional DNN model and the Kalman-CNN model, respectively. Second, this chapter further studied their performances as shown in Fig. 3.8b. Overall, Kalman-based deep learning models achieve better localization results with smaller errors than conventional deep learning models. For instance, the Kalman-CNN improves the medium localization error by 0.2 $m$ in comparison with conventional CNN model. It seems the Kalman-CNN performs poorer than the Kalman-DNN because its parameter



**(a)** CDF

**(b)** Box-plot

**Figure 3.8 .** The overall performance comparisons among Kalman-DNN and the baseline methods.

complexity is not enough to realize more useful feature learning for localization.

The above experimental results show that the Kalman filters can effectively tame the noises in RSS time-series and improve localization accuracy. By exploiting temporal features of RSS measurements, the proposed Kalman-DNN model achieves the best performance among all other methods in the literature.

## 3.5 Conclusion

In this section, this chapter proposed a novel indoor localization method utilizing the temporal dependency of RSS data with DNN for IoT-oriented wireless indoor localization. The Kalman filter-based RSS filtering algorithm is leveraged to tame the random noises in RSS time-series data. To further reduce the computation complexity, a post-processing module has been proposed together with a label processing module. To efficiently extract robust features, a DNN-based deep learning model is further applied with online training process. Extensive field experiments have been conducted using a real-world testbed, and experimental results validate the effectiveness of the proposed Kalman-DNN model. Overall, the Kalman-DNN model can improve up to 25% localization accuracy in comparison with the conventional DNN model. In addition, the Kalman-DNN model outperforms the Kalman-CNN model by 10% in localization accuracy with the mean localization time of 0.02 $ms$.

# Chapter 4

# CapsLoc: A Robust Indoor Localization System with Wi-Fi Fingerprinting using Capsule Networks

With the unprecedented demand of location-based services in indoor scenarios, wireless indoor localization is emerging as an essential technology for mobile users. While line-of-sight Global Positioning System (GPS) signal is not available in indoor space, Wi-Fi fingerprinting using RSS has become popular with its ubiquitous accessibility. Although fingerprinting data can be easily collected by portable mobile devices, how to achieve robust and efficient indoor localization remains challenging with two constraints.

First, localization accuracy is degraded by random fluctuation of signals that comes from multipath effects and noise of RSS signals and environment dynamic. Second, indoor localization algorithms are time-consuming due to the handcrafting features and complex filtering on raw data. To achieve high localization accuracy with Wi-Fi fingerprinting, this chapter proposes CapsLoc, a robust indoor localization system based on capsule networks. Specifically, capsule network model can efficiently extract hierarchical structures from Wi-Fi fingerprint with three main components: a convolutional layer, a primary capsule layer and a feature capsule layer. This chapter conducts real-world experiments to test over 33600 data points. The experimental results show that CapsLoc can achieve accurate indoor localization with an average error of $0.68m$, which outperforms conventional machine learning methods (KNN and SVM) and existing deep learning methods (CNN and SAE-CNN).

## 4.1 Introduction

localization technology has played a significant role in the era of IoT. With ubiquitous deployment of wireless systems and pervasive use of smart devices, ILBSs, such as marketing and advertising [48], tracking and navigation [49], interactive and personalized routing [50], have become essential for smart cities. While GPS signal is too sensitive to occlusion and it cannot deliver satisfactory ILBSs, many different wireless technologies [42] have been utilized in the literature, including Wi-Fi, Bluetooth, RFID, UWB, radar and cellular networks, *etc.* Among the above, Wi-Fi has the widest availability and is supported by the majority of mobile devices, thus it becomes the most popular radio signal-based technique for indoor localization. A variety of measurements have been proposed to enable Wi-Fi-based indoor positioning, including AOA, TOA, CSI and RSS [42]. In recent years, the RSS-based fingerprinting for localization has received numerous efforts to achieve accurate and practical Non-Line-of-Sight localization.

The basic rationale of Wi-Fi fingerprinting for localization is that indoor location can be identified by a record of RSS signatures from surrounding Wi-Fi APs. Accordingly, fingerprinting-based localization generally consists of two phases [51]: (1) an offline training phase for RSS fingerprint collection and localization model construction; (2) an online phase for real-time localization by RSS-location mapping.

For the offline phase, fingerprints are usually collected at evenly distributed reference points (RPs) across indoor space. While for the online phase, localization systems employ localization algorithms to find the best match between the fingerprint and indoor location.

To enable accurate and real-time online localization, a key challenge is to extract useful and reliable features from fingerprint database, and further develop effective mapping functions to perform localization. In the past, most of conventional RSS fingerprint-based localization systems have relied on machine learning algorithms such as KNN, SVM and Compressive Sensing (CS) [52]. However, these methods are subject to variations and fluctuations of RSS signals as they can only learn handcrafted features from fingerprinting data. Furthermore, conventional machine learning algorithms only have shallow representation spaces, and they typically cannot attain refined features from floor-level and building-level fingerprinting datasets, which may have up to hundreds of Wi-Fi APs involved.

To efficiently extract high-dimensional representation from complex fingerprint data, neural network-based architectures have been proposed recently. The core idea of deep learning is to learn features from data in an incremental, layer-by-layer way and jointly develop complex representations. Existing deep learning architectures for indoor localization include DNN, Deep Belief Network (DBN), CNN, SAE and Recurrent Neural Network (RNN) [53]. Among these architectures, CNN shows remarkable performance in processing data in the form of arrays by extracting high-level features with consecutive convolution operations and pooling operations. However, some valuable spatial information of layer-to-layer neurons may get lost with pooling operations (*e.g.*, max pooling) in CNNs. For indoor localization, losing such information can directly degrade localization accuracy, as RSS-based fingerprints are spatially distributed and have strong correlations. To address above problem, Hinton *et al.* [36] proposed CapsNet as an alternative to CNNs. The Capsules are composed of neurons that use vectors to learn and store feature information, with each neuron's output representing a different property of each feature. Consequently, CapsNet is efficient in capturing hierarchical structure of entities in input data such as images. For example, the CapsNet model trained from scratch on multi-MNIST training data achieves higher test classification accuracy than the convolutional baseline model [36]. Indoor localization with RSS fingerprinting can also be considered as a classification problem.

To enhance robustness and accuracy of indoor localization, this chapter proposes CapsLoc, an indoor localization system by using CapsNet with Wi-Fi fingerprinting data. In particular, this chapter builds a real-world experimental system of CapsLoc to test the performance of CapsNet for indoor localization. The experimental results show that CapsLoc can achieve accurate indoor localization with an average error of $0.68m$, which outperforms conventional machine learning methods (KNN and SVM) and existing deep learning methods (CNN and SAE-CNN).

The remainder of this Chapter is organized as follows. In Section 4.2, this chapter provides a literature review of related work on RSS-based indoor localization and applications based on CapsNet. Next, this chapter introduces CapsLoc system in data processing, offline training and online localization in Section 4.4. Then, this chapter presents the design of using CapsNet for robust indoor localization in Section 4.3. Section 4.5 is about the experimental performance analysis. Finally, this chapter is summarized in Section 4.6.

## 4.2 Related Work

### 4.2.1 Review of Indoor Localization with RSS Fingerprinting

With ubiquitous accessibility of Wi-Fi APs in indoor space, RSS fingerprinting has become one of the most promising methodologies for indoor localization. Meanwhile, there are still some key challenges [52], such as multipath effects, fluctuation of RSS signals and the trade-off between data collection and localization accuracy, which need to be formally addressed when performing localization with RSS fingerprinting. To achieve higher localization accuracy, a variety of machine learning methods have been developed for indoor localization. For instance, Li *et al.* [54] proposed a feature-scaling-based KNN algorithm to assign differential weights to signal differences at different RSS levels with improved localization accuracy. Hong *et al.* [55] proposed a device-free passive localization with RSS signals by using multi-class SVMs to process a combination of spatial and temporal array signal features.

In an early attempt of deep learning for indoor localization, Moreover, Zhang *et al.* [56] tamed the variant and unpredictable RSS signals for positioning with a four-layer DNN, which is pre-trained by Stacked Denoising Autoencoder to learn reliable features from noisy samples without hand-engineering. Song *et al.* [6] proposed a novel CNN combined with SAE to deliver more accurate floor-level localization, with the model scalable to different indoor environments and datasets. Recently, RNN and Long Short-Term Memory models have also been adopted to perform indoor localization with sequential dataset to enhance the localization accuracy in large-scale indoor spaces [57].

### 4.2.2 Capsule Networks

The concept of 'capsules' was firstly introduced by Hinton *et al.* [58]. Sabour *et al.* [36] proposed CapsNet with dynamic routing for Capsules. Since then, many innovative models have been proposed based on CapsNet for different applications, including feature representation [59], image classification [60], audio processing [61] and multi-task learning [62]. In [63], Own *et al.* used SVM model in indoor environments and further employed conventional capsule networks to process 2.4G and 5G Wi-Fi signals. In this work, this chapter proposes CapsNet-based solution for robust and accurate indoor localization with Wi-Fi fingerprinting. The experimental results validate the effectiveness of CapsNet in extracting high-level features from Wi-Fi fingerprinting.

## 4.3 Capsule Networks

The core architecture for CapsNet is shown in Fig. 4.1. The input data is the preprocessed data as introduced in Subsection 4.4.1. Then, the data is processed by the convolutional operation with filters in the Convolutional (Conv) Layer. In the next Primary-Capsule (PC) Layer, the data is further processed by a Conv with squash activation and then is reshaped. After that, the processing of reshaped data is routed based on 'Dynamic Routing' scheme to derive feature capsules in Feature-Capsule (FC) Layer. At last, an Auxiliary Layer replaces each capsule in FC Layer with its length matching the true label in form of One-Hot encoder [47].

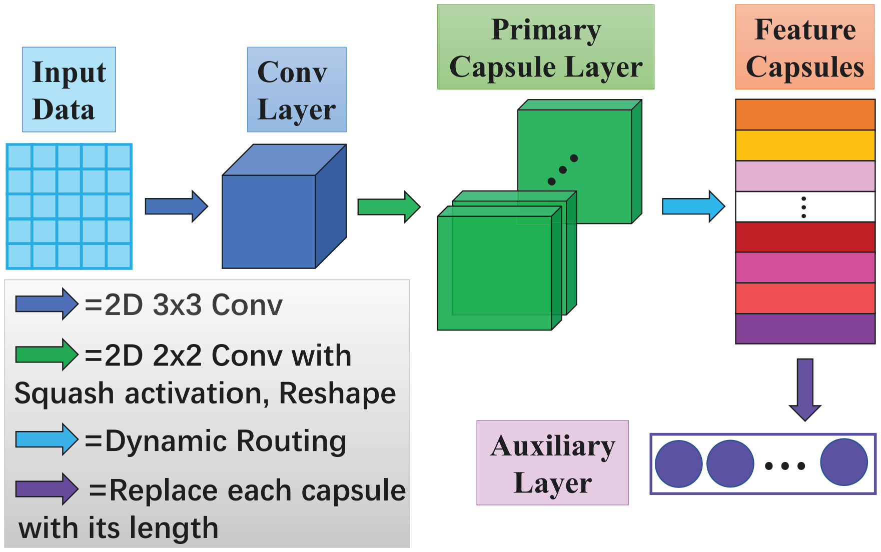


**Figure 4.1 .** The architecture of CapsNet for indoor localization

This chapter introduces core operations in the CapsNet as follows.

### 4.3.1 Convolution Operation

Let $x_i \in R$ be one-dimensional data. The input data vector $\boldsymbol{x}^j$ of length $n$ is represented as $\boldsymbol{x}^j = [x_1^j, x_2^j, \ldots, x_n^j], j = 1, 2, \ldots M$, where $M$ is the number of training points and $n$ is the number of APs. A convolution operation involves a filter $\boldsymbol{w}^j \in \mathbb{R}^n$, which is applied to the vector $\boldsymbol{x}^j$ to produce a new feature. For instance, a feature $c_i^j$ is generated from the vector $\boldsymbol{x}^j$ by:

$$c_i^j = f(\boldsymbol{w}^j \,\boldsymbol{.}\, \boldsymbol{x}^j + b^j). \tag{4.1}$$

Here, notation $\boldsymbol{.}$ is the convolution operation. $b^j \in \mathbb{R}^n$ is a bias term; $f$ is a non-linear activation function that introduces nonlinearities to CNN and is an ideal method for multilayer networks to recognize the nonlinear characteristics of input data. The filter $\boldsymbol{w}^j$ is applied to every vector $\boldsymbol{x}^j$, where $j$ is the number of the vector to produce a $featuremap$ as[36]:

$$\boldsymbol{c}^j = [c_1^j, c_2^j, \dots c_n^j]. \tag{4.2}$$

### 4.3.2 Dynamic Routing

A capsule is defined as a group of neurons in a CapsNet. It is a vector that has both direction and length. The direction of capsule captures the entity's attributes. The length of capsule represents the probability of entity existence. The shortcomings of CNNs mainly stem from pooling layers. As in CapsNet, these layers are replaced with a more appropriate standard, namely 'routing by agreement'. According to this standard, the outputs are sent to all parent capsules in the layer below. However, their coupling coefficients are not equal. Each capsule makes an effort to measure the output of its parent capsules. Once the prediction corresponds to the practical output of a parent capsule, coupling coefficients between those two capsules are increased.

$\boldsymbol{u}_i$ is taken as the output of capsule $i$, the prediction of parent capsule $j$ is calculated as:

$$\widehat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i, \tag{4.3}$$

where $\widehat{\boldsymbol{u}}_{j|i}$ is the prediction variable of the output of the $j$th capsule at higher level, calculated by capsule $i$ of the PC layer; and $\boldsymbol{W}_{ij}$ is the weighting matrix to be learned by the CapsNet in backward pass. According to the degree of conformation between the capsules of the layer below and parent capsules, coupling coefficients $c_{ij}$ can be calculated according to the following Softmax function:

$$c_{ij} = \frac{exp(b_{ij})}{\sum_j exp(b_{ij})}, \tag{4.4}$$

where $b_{ij}$ is the logarithmic probability that whether capsule $i$ is coupled with capsule $j$, and it is given as 0 at the early stage of the process, namely routing by agreement. Accordingly, the input vector for the parent capsule $j$ is calculated as:

$$\boldsymbol{S}_j = \sum_i c_{ij} \widehat{\boldsymbol{u}}_{j|i}. \tag{4.5}$$

At last, the non-linear squashing function below is utilized to prevent the output vectors of Capsules from exceeding, through which the final output of each Capsule is formed according to its initial vector' value. The function now is defined in Equation 4.5:

$$\boldsymbol{v}_j = \frac{||\boldsymbol{S}_j||^2}{1 + ||\boldsymbol{S}_j||^2} \frac{\boldsymbol{S}_j}{||\boldsymbol{S}_j||}, \tag{4.6}$$

where $\boldsymbol{S}_j$ is the input vector of Capsule $j$ and $\boldsymbol{v}_j$ is the output; "||" means logical disjunction operation. The log probabilities should be iterated in the routing process according to the agreement between $\boldsymbol{v}_j$ and $\widehat{\boldsymbol{u}}_{j|i}$, based on the fact that once the two vectors agree, there is a large inner product. So the agreement $a_{ij}$ that is used to iterate log probabilities $b_{ij}$ and coupling coefficients $c_{ij}$, is computed as:

$$a_{ij} = \boldsymbol{v}_j \widehat{\boldsymbol{u}}_{j|i}. \tag{4.7}$$

Each capsule $k$ in FC layer is related to a loss function $l_k$, which puts a high loss value on capsules with long output instantiation parameters when the entity does not actually exist. The loss function $l_k$ is defined as:

$$l_k = T_k max(0, m^+ - ||\boldsymbol{v}_k||)^2 + \lambda(1 - T_k) max(0, ||\boldsymbol{v}_k|| - m^-)^2, \tag{4.8}$$

where $T_k$ is 1 whenever class $k$ is actually present and is 0 otherwise. Terms $m^+$, $m^-$, and $\lambda$ are the hyper parameters to be learned in the training process[36].

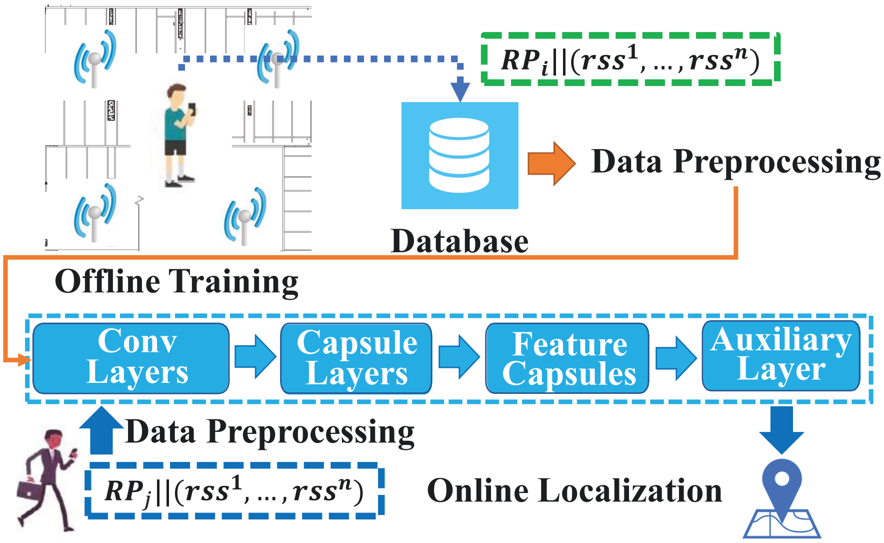


**Figure 4.2 .** The framework of CapsLoc indoor localization system

## 4.4 Overview of the CapsLoc

This section presents an overview of the proposed Capsule Network based indoor localization for IoT devices. Fig. 4.2, shows the architecture of CapsLoc, as this chapter utilizes signals from Wi-Fi APs in an indoor environment to create the RSS fingerprint. This chapter assumes an indoor scenario with $n$ APs where there are $k$ RPs across the entire space. At each RP, this chapter samples $m$ fingerprints that are labeled by the two-dimension location (row and column) on the ground. In this way, the unrolled fingerprint database can be regarded as a huge matrix containing $mk * n$ vectors, where $mk = m * k$; notation $*$ is the multiplication operation. Then, in the offline phase, training data from database is firstly pre-processed by the data processing module and then further utilized to train CapsNet model. In the online localization phase, testing data will also be fed into the data processing model and then the trained CapsNet model to estimate the locations of mobile users.

### 4.4.1 Data Preprocessing

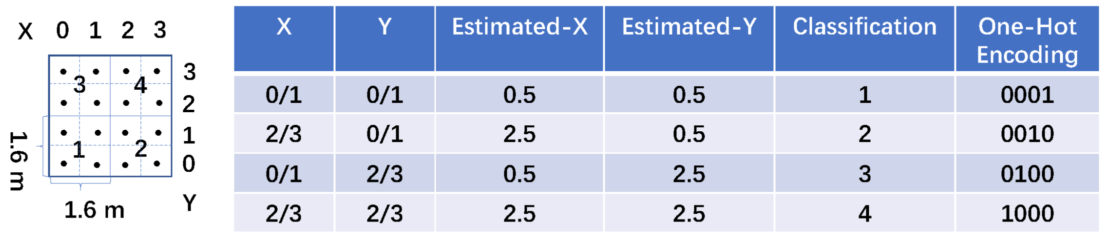
1) RSS Data Processing. To enrich the representations of RSS data, this chapter increases the dimension of original RSS fingerprints by adding a new feature set. The detailed features are set as follows [64]:

•*Raw*: The original features that are directly generated from the RSS fingerprints.

•*R*: A set of features that represent the mutual differences of RSS values from different APs. For instance, a basic entry of $r_i - r_j$, where $i, j \in n$, can represent the difference between the RSS value from AP $i$ and AP $j$. Entirely, the feature arrays of each individual AP together form a feature matrix $R$ as:

$$\boldsymbol{R} = \begin{bmatrix} 0 & r_1 - r_2 & r_1 - r_3 & \cdots & r_1 - r_n \\ r_2 - r_1 & 0 & r_2 - r_3 & \cdots & r_2 - r_n \\ r_3 - r_1 & r_3 - r_2 & 0 & \cdots & r_3 - r_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_n - r_1 & r_n - r_2 & r_n - r_3 & \cdots & 0 \end{bmatrix}. \tag{4.9}$$

2) Label Preprocessing. To determine the label of RSS fingerprints at each reference point, this chapter first divides the localization area into a number of small zones, *i.e.*, where each zone is a grid area covering $1.6 \times 1.6\ m^2$. To generate the label for each grid, this chapter adopts One-Hot Encoding [47] to map each grid onto a One-Hot vector. Consequently, each individual grid represents as a categorical variable, and then indoor localization task essentially becomes a classification problem across all grids with the fingerprints. For example, as shown in Fig. 4.3,



| X | Y | Estimated-X | Estimated-Y | Classification | One-Hot Encoding |
|---|---|---|---|---|---|
| 0/1 | 0/1 | 0.5 | 0.5 | 1 | 0001 |
| 2/3 | 0/1 | 2.5 | 0.5 | 2 | 0010 |
| 0/1 | 2/3 | 0.5 | 2.5 | 3 | 0100 |
| 2/3 | 2/3 | 2.5 | 2.5 | 4 | 1000 |

**Figure 4.3 .** The label processing example for BUPT database.

when there are 16 reference points (even-numbered), the labeling process works as below: In the training phase on the edge server, these reference points are classified into 4 types and are further mapped onto 4 one-hot encodings, as shown in the Table. In turn, in the localization phase at edge devices, one-hot encodings generated by the CapsNet model will be decoded into corresponding X/Y coordinates, so as to estimate the actual user location.

### 4.4.2 Offline Training Phase

The preprocessing results of $N$ training samples are further utilized to train the CapsNet model. After training session, the trained CapsNet model will be capable of localizing mobile users with their fingerprints as the inputs. Such that the proposed CapsLoc system can perform real-time positioning during the online localization phase.

### 4.4.3 Online Localization Phase

The online phase is to provide real-time indoor localization for mobile users. A mobile user with unknown location can simply send current measurement of RSS fingerprint to the CapsLoc system. Then, the real-time data will be preprocessed and fed into the trained CapsNet model to determine the most possible grid that the mobile user belongs to. In this way, the CapsLoc system can use the center of the corresponding grid as the estimated user location.

## 4.5 Experiment Setup and Evaluation

### 4.5.1 Experimental Setup

This chapter implements a real-world indoor localization system of CapsLoc in the IoT lab at Beijing University of Posts and Telecommunications as illustrated in Fig. 3.5. To achieve efficient localization, this chapter deploys 6 Wi-Fi APs to cover an area of $460m^2$, where there are three lab rooms (each room has two APs) along with a corridor area. Here, the APs are TP-Link wireless routers. Meanwhile, this chapter installs a number of RPs that are evenly distributed across the space, with distance of $0.8m$ between two adjacent RPs.

### 4.5.2 Dataset Collection

In real-world indoor space, multi-path effects and fluctuations of RSS signals always challenge accuracy and stability of indoor localization. In CapsLoc system, this chapter employs a laptop installed with Phoenix Wi-Fi Collector to collect (software for collecting RSS data) and store Wi-Fi fingerprints. To tame the variations in RSS signals, this chapter samples the RSS fingerprints of 6 APs at each RP for 300 times. These samples are stored in a local database for training and testing purposes. Overall, this chapter collected 33600 data points, with each data point labeled by the location of row and column shown in Fig. 3.5. Thereafter, this chapter splits the dataset into two parts: 60% data points for training and 40% data points for testing.

### 4.5.3 Baseline Methods

To validate the performance of CapsNet for indoor localization, this chapter further employs some conventional machine learning methods and existing deep learning methods as baselines. Particularly, this chapter performs the same localization experiments of CapsNet with CNNLoc [5], CNN [65], KNN [54] and SVM [19]. This chapter implements CapsNet along with the above baseline methods using framework of Tensorflow and Keras in Python 3.6. The hardware platform is a DELL Latitude 5480 equipped with a 4-thread Intel i7-7600U CPU of 2.9 GHz and 16GB RAM.

### 4.5.4 Evaluation Studies

**Experiment evaluation indicators.** The experiment evaluation indicators of this chapter include localization error, localization accuracy, mean positioning time. Localization error is the Euclidean distance between estimated coordinates and actual ones of the same location. Localization accuracy is the probability that estimated location classification of testing data is correct. Positioning time is the time that one testing data is processed by all online modules. Mean positioning time is the mean time of all testing data.

**Parameter Settings:** The basic dataflow of CapsNet for indoor localization is depicted in Fig. 4.4, where the CapsNet in CapsLoc consists of five layers: an input layer, a convolution layer, a primary capsule layer, a digit capsule layer and

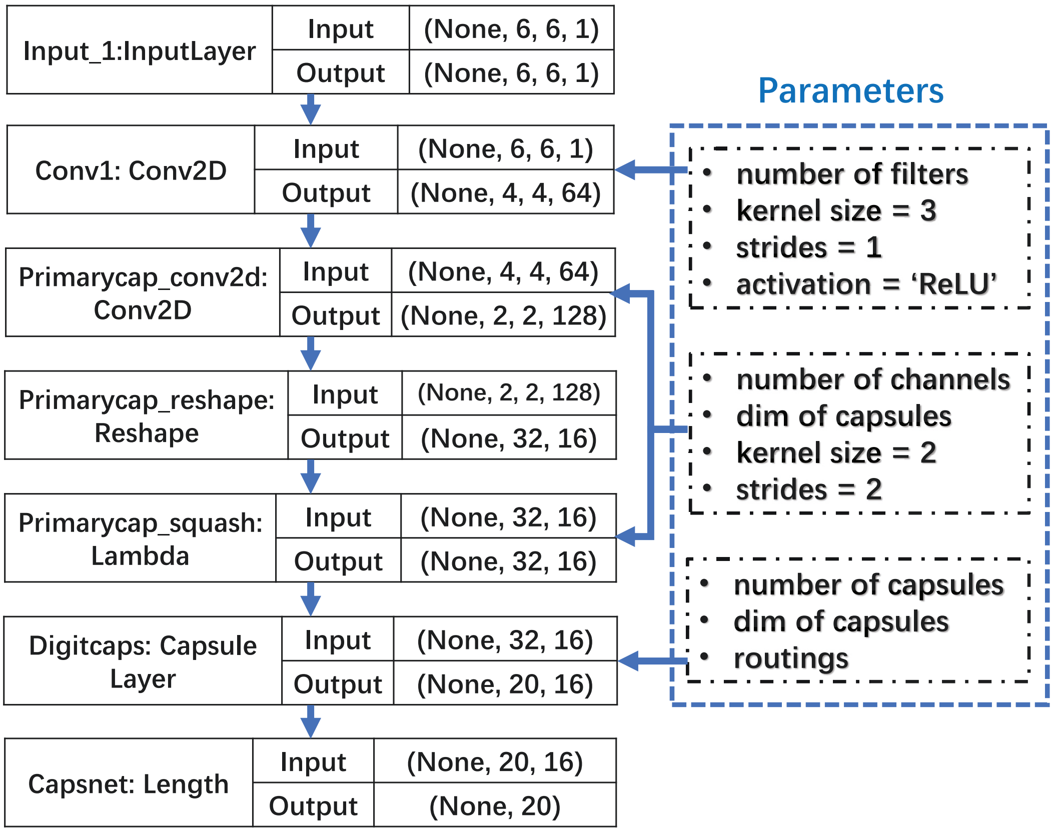


**Figure 4.4 .** The graph of layers and dataflow in CapsNet for indoor localization

an output layer. The input and output data of each layer are all specified in tensor format, and the input and any layer share the tensor with the same shape. The basic parameters of each individual layer are listed in the Parameters Blocks. For Conv1, this chapter sets the convolutional kernel size as 3, convolutional strides as 1, 'ReLU' as the activation function. This chapter further evaluates the impact of *number of filters (n_filters)*. For Primarycap, this chapter sets the convolutional kernel size as 2, convolutional strides as 2, 'Squash' as the activation function, and further evaluate the impact of *number of channels (n_channels)* and *dimension of capsule (dim_capsule)*. For Digitcaps (the feature layer), this chapter sets the number of routing iterations as 3 and number of capsules as the number of grids of the area, and further evaluate the impact of *dim_capsule*. Note that, this chapter sets the same number of *dim_capsule* for both Primarycap and Digitcaps.

**Baseline Comparison:** First, the distribution of localization errors by different methods is shown in Fig. 4.5. KNN has low localization performance where more than 60% testing results have errors ranging from $2m$ to $8m$. Meanwhile, SVM-based localization method has some improvement, but still 20% testing results are with errors larger than $2m$, and 5% of the testing cases have errors of $3m$ and more. In addition, CNN-based localization methods can extract high-level features and further enhance the localization accuracy, with 80% to 90% testing samples having errors less than $2m$, respectively. It can be seen that SAE-CNN improves the performance of CNN by encoding raw RSS fingerprints into features as the input of CNN. Since CapsNet preserves the valuable spatial information for between-layer neurons, it successfully achieves the best indoor localization results over all baseline methods, with 99% testing results being within errors lower than $2m$ and over 40% testing results being within errors around $1m$. To make it clearer, this chapter shows the results by a box-plot in Fig. 4.6. It is obvious that KNN method has the biggest distance between maximum error and minimum error as well as the largest Interquartile Range (IQR, *i.e.*, the distance between first quartile and third quartile). In comparison with CNN and SAE-CNN, CapsNet has slightly smaller IQR and a much lower average localization error of $0.68m$. The above results also show that CapsNet outperforms both conventional machine learning methods and state-of-the-art deep learning methods for the indoor localization system of CapsLoc. Next, this chapter evaluates the impact of parameter settings of CapsNet on localization

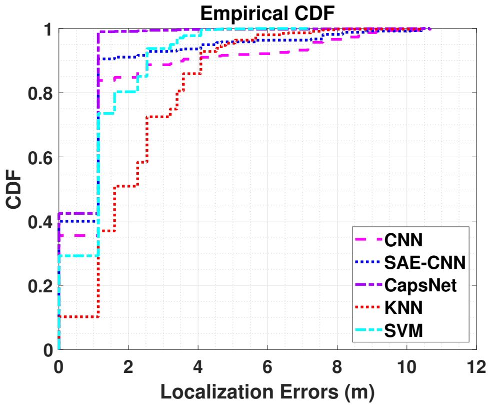


**Figure 4.5 .** The CDF distribution of localization errors

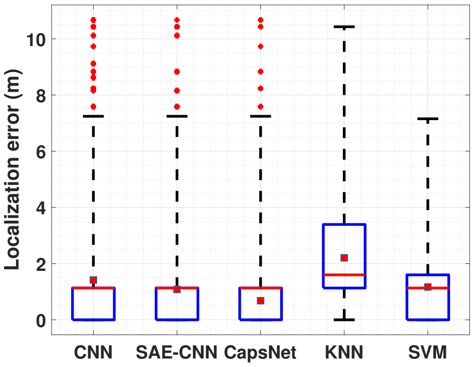
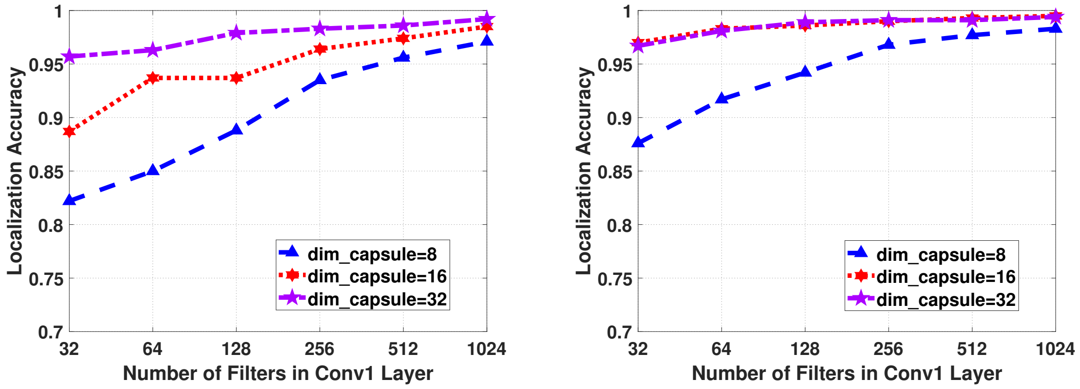


**Figure 4.6 .** The box-plot distribution of localization errors

accuracy and positioning time.

**Parameter's impact on localization accuracy:** As this chapter has mentioned in parameter settings part, this chapter firstly shows how different parameters in CapsNet impact localization accuracy in CapsLoc. For the Primarycap layer, this chapter sets the *n_channels* as 8 and 16 for evaluation purposes. In the meantime, this chapter tunes the *dim_capsule* (*i.e.*, the dimension of the output vectors by the capsules) from 8 to 16 and 32. Then, this chapter conducts a grid search over 32, 64, 128, 256, 512, 1024 for the *n_filters* in Conv1 layer. The evaluation results of parameter's impact on localization accuracy are presented in Fig. 4.7. First, overall localization accuracy improves stably with larger *n_filters* in Conv1 layer. Meanwhile, as the *n_channels* increases, localization accuracy has a slight enhancement. Second, the *dim_capsule* has a direct impact on the indoor localization accuracy, since that CapsNet shows up to 10% accuracy improvement in both subfigures when the *dim_capsule* is 8 and 32, respectively. However, with the increasing *n_filters* in



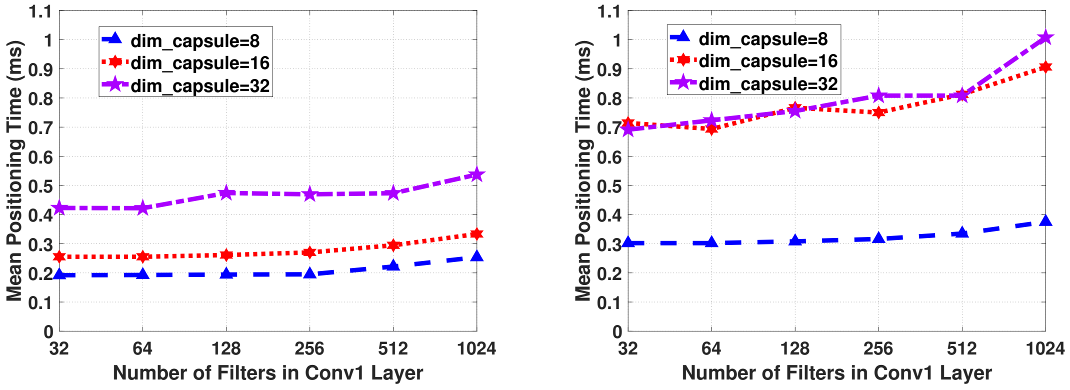
**(a)** Primarycap layer with 8 channels

**(b)** Primarycap layer with 16 channels

**Figure 4.7 .** Comparison of localization accuracy by CapsLoc with 8 channels and 16 channels

Conv1 layer, such gap generally shrinks to below 5%. The above evaluations give us an insight of contributions of different components in CapsNet, while the trade-off between localization accuracy and parameter settings is further investigated in the following.

**Parameter's impact on mean positioning time:** Similar to the scenarios of evaluations on localization accuracy, this chapter sets the *n_channels* of Primarycap layer as 8 and 16, respectively, with other parameters using the same tuning space. Since the positioning time would directly influence the user experience of an indoor localization system, in this section, this chapter explores the impact of various parameters on the mean positioning time of CapsNet. As shown in Fig. 4.8a and Fig. 4.8b, positioning time generally increases with increased *n_channels* in Primarycap layer, *dim_capsule* and *n_filters* in Conv1 layer. Therefore, by jointly considering the performance on localization accuracy in Fig. 4.7, this chapter finds



**(a)** Primarycap layer with 8 channels

**(b)** Primarycap layer with 16 channels

**Figure 4.8 .** Comparison of positioning time by CapsLoc with 8 channels and 16 channels

a CapsNet with 512 filters in Conv1 layer, 32 capsules and the *dim_capsule* of 32, can achieve the best trade-off between localization accuracy and positioning time (*e.g.*, 98% accuracy within an average time of 0.5 ms).

## 4.6 Summary

This chapter proposed the CapsLoc, a robust indoor localization system with Wi-Fi fingerprinting using CapsNet, which can overcome the typical challenges. The experimental results validate the effectiveness of CapsNet, which outperforms the CNN-based indoor localization methods and conventional KNN-based and SVM-based methodologies. With extensive experimental results, it is believed to be worthwhile to preserve spatial hierarchies in extracting high-level features from RSS fingerprint dataset for high-accuracy indoor localization.

# Chapter 5

# EdgeLoc: Capsule Network based Indoor Localization towards Heterogeneous IoT Edge Devices

Indoor localization has become an essential demand driven by ILBSs for mobile users. With the rising of IoT applications and services, heterogeneous smartphones and wearables have become ubiquitous. However, ILBSs for heterogeneous IoT devices confront significant challenges, such as RSS variances caused by the superimposed factors (hardware heterogeneity, energy constraint, and database related problem, which is described in Section1.2), and other classic factors with wireless communications (multipath reflections and environment dynamics).

In this Chapter, EdgeLoc is presented, which is a robust and real-time indoor localization system for heterogeneous IoT devices by solving the above challenges. In particular, RSS fingerprinting data of Wi-Fi is employed for localization by tackling the heterogeneity of IoT devices in two folds: first, feature-level and signal-level solutions are presented to address the problem of random RSS variances; at the feature level, this chapter proposes a novel capsule neural network model to efficiently extract incremental features from RSS fingerprinting data. At the signal level, a multi-step data-flow is further devised to convert RSS fingerprints to image-like data, which utilizes feature matrix to reduce absolute sensing errors introduced by device heterogeneity. Second, an edge-IoT framework is designed to utilize edge server to train deep learning model and further support real-time localization for heterogeneous IoT devices. Extensive field experiments with over 33,600 data points are conducted to validate the effectiveness of EdgeLoc with a large-scale Wi-Fi fingerprint dataset. The results show that EdgeLoc outperforms the existing SAE-CNN method in localization accuracy by up to 14.4%, with an average error of 0.68 $m$ and an average positioning time of 2.05 $ms$.

## 5.1 Introduction

With the ubiquitous deployment of Wi-Fi networks and pervasive use of portable smart devices, ILBSs, such as mobile advertising [66], navigation [67], and interactive routing [68], are prevailing as part of smart city solutions. As a fundamental prerequisite for ILBSs, indoor localization is considered as a significant necessity. Since GPS signal is susceptible to occlusion (*e.g.*, buildings), it cannot provide indoor localization service. In contrast, Wi-Fi networks are widely available in smart buildings and supported by smart devices, thereby has become a promising technology to enable indoor localization [30]. Recently, considerable efforts have been made in Wi-Fi-based signal strength fingerprinting (RSS) to achieve NLOS indoor localization. Intuitively, it is generally assumed that an indoor location can be identified based on a unique signal vector of RSS values measured from different Wi-Fi APs [52]. Fingerprinting-based localization generally consists of two phases [51]: (1) an offline phase for data collection, fingerprint database construction and model training; (2) an online phase for indoor localization through mapping users onto a radio map [42].

While conventional RSS fingerprint-based approaches use the same type of devices for training and testing models, heterogeneity of IoT devices poses significant challenges to indoor localization [69]. With different signal measuring sensitivities, RSS values measured from heterogeneous IoT devices fluctuate even at the same locations, thereby causing mismatch problems in cross-device fingerprinting database [70]. In other words, random variances of RSS values are caused not only by multipath effects from complex indoor environments [71], but also by different hardware specifications of IoT devices. In addition, computation capacity varies from device to device in heterogeneous IoT systems, which makes it hard to guarantee real-time calculation of localization. Consequently, both model training (offline) and real-time localization (online) phases of fingerprint-based localization are influenced, leading to unfavorable errors and positioning delays in localization service.

To this end, two major challenges need to be addressed for indoor localization with IoT device heterogeneity.

- **Challenge 1**: *Taming cross-device fingerprint with random variances*: Multipath effect of Wi-Fi signals is ubiquitous in indoor scenarios [72], which ultimately causes random fluctuations of RSS. In the meantime, RSS variances caused by IoT heterogeneity need to be further eliminated. While existing work has proposed signal-level calibrations for RSS measurements [73–75], the requirement of localization accuracy calls for more comprehensive solutions to make cross-device fingerprint compatible.

- **Challenge 2**: *Locating heterogeneous devices in real-time*: ILBSs have rigorous instantaneity requirements of localization [76–78]. However, the difficulty of delivering real-time localization is increased by computation capacity difference among heterogeneous devices. Deep neural network models generally have tens of thousands of or millions of parameters to learn, thus it becomes much more daunting to achieve deep learning-based indoor localization with heterogeneous devices in real-time.

To tackle the above challenges, EdgeLoc, a robust and real-time indoor localization system towards heterogeneous edge devices of IoT, is designed and implemented. Firstly, in Challenge 1, RSS data from heterogeneous devices are processed at both feature level and signal level. At the feature level, a deep learning model is leveraged to automatically extract multi-level features from cross-device fingerprinting data. In specific, noises in cross-device fingerprints are tamed by utilizing a CapsNet to learn high-dimensional representations from complex fingerprint data. While deep learning-based CNN shows remarkable performance in extracting high-level features, some valuable spatial information of layer-to-layer neurons may be lost through pooling operations [79]. For cross-device fingerprints, missing such information can directly degrade localization accuracy, as RSS-based fingerprints are spatially distributed and have strong correlations [2]. Therefore, a novel CapsNet is designed to efficiently capture the hierarchical features in cross-device RSS fingerprinting data. In this CapsNet model, capsules are composed of neurons that learn and store feature information, with each neuron's output representing the unique property of the same feature [58]. At the signal level, a two-dimensional Signal Strength Difference (SSD) in feature matrix is devised to represent mutual differences among RSS measurements from different APs. Rather than utilizing the original RSS as a location fingerprint, SSD provides a more stable location signature for any mobile device irrespective of its hardware specification [80]. In addition, the differences in feature matrix would expand feature dimension and further facilitate the CapsNet model to learn valuable features.

Secondly, to address the second challenge above, a hierarchical edge-IoT architecture is proposed to utilize computation resource of an edge server to meet real-time localization requirement. An edge server has a local database to store RSS data and trained localization model [81]. Once a mobile device senses its RSS fingerprint, it can acquire optimized model parameters from an edge server and leverage the well-trained model to compute its own location in real-time [82].

The contributions of this work are summarized as follows:

- EdgeLoc, a real-time and high-accurate localization system taking heterogeneous IoT devices into consideration is proposed. To the best of the knowledge, this is the first study to combine edge computing with CapsNets to tame device heterogeneity of indoor localization.

- The problem of random variances of RSS data collected at heterogeneous devices is solved at both signal level and feature level to find the best CapsNet model with training data to maintain robustness and accuracy of indoor localization. At the feature level, a capsule neural network-based deep learning model is proposed to efficiently capture the hierarchical representations from cross-device fingerprinting data. At the signal level, a two-dimensional feature matrix is constructed with mutual SSD as the input of CapsNet.

- The method leverages the edge-IoT framework to build a real-world indoor localization system. The architecture of edge-IoT framework is hierarchical, which consists of edge server, edge devices, and Wi-Fi APs for real-time indoor localization. In particular, an edge server performs model training and offloads optimized parameters to heterogeneous devices. Therefore, heterogeneous IoT devices are free from computation-costly model training and can only focus on real-time localization with one-hop communication delay and low computation complexity.

- Extensive evaluations are conducted on the EdgeLoc system with a real-world field experiment and a large-scale fingerprinting dataset (*i.e.*, UJIIndoorLoc dataset). The results demonstrate that EdgeLoc achieves 98.5% localization accuracy with heterogeneous IoT devices by combining edge computing with deep learning, with an average error of 0.68 m and an average positioning time of 2.05 ms.

The rest of this Chapter is organized as follows: Section 5.2 provides state-of-the-art literature review on edge computing, indoor localization and capsule networks. The architecture and dataflow of the EdgeLoc system are presented in Section 5.3 and Section 5.4, respectively. Section 5.5 presents the experimental results from implemented EdgeLoc system with comprehensive analysis. Finally, Section 5.6 concludes this Chapter.

## 5.2 Related Work

In this section, the related work on indoor localization is discussed including classic wireless indoor localization methods and RSS-based Wi-Fi fingerprinting methods.

**Wireless Indoor Localization.** Indoor localization has received numerous research efforts in the past decades and can be divided into two sub-domains: device-based localization and device-free localization [42]. In device-based localization, a target equipped with a wireless device (*e.g.*, a smartphone) can be located through wireless signals sent from other terminals (*e.g.*, Wi-Fi APs) [83]. In contrast, device-free localization methodologies are less intrusive and can identify locations of target entities via their signal reflections. Overall, various wireless signals support both device-based and device-free indoor localization [51], including Wi-Fi, acoustics, FM radios, Bluetooth, cellular signals, UWB radar, RFID, and LoRa *etc.* Wi-Fi-based indoor localization is the most ubiquitous approach and Wi-Fi technology provides multiple location-related measurements, such as TOA, AOA, RSS and CSI [84]. The work utilizes the RSS for indoor localization in a novel way that image-like fingerprints are created to enrich the representations of original input data. Moreover, this chapter applies edge computing method in RSS-based indoor localization for efficient location computing with a massive amount of fingerprinting data.

**RSS-based Wi-Fi Fingerprinting for Indoor localization.** The received signal strength of Wi-Fi has the advantage of easy accessibility, NLOS and low cost, making it feasible to enable wireless indoor localization [24]. With the intensive deployment of Wi-Fi networks and the complex indoor environment, multi-path effect, device heterogeneity and signal variation can significantly influence RSS and degrade the performance of conventional RSS-based distance estimation for indoor localization [43]. Consequently, the RSS-based fingerprinting for indoor localization is proposed by leveraging a unique signal vector of RSS values as the label for each location [43]. The basic procedure of indoor localization with Wi-Fi RSS fingerprints consists of an offline phase to establish radio maps and an online phase to map the RSS measurement onto a corresponding location in the radio map. Various machine learning-based and deep learning-based methods have been developed for RSS-based indoor localization [6, 64, 85]. For instance, Felix *et al.* used DNN and DBN to reduce errors in the dynamic indoor environment for localization [85]. Moreover, Song *et al.* proposed a scalable neural network model by combining CNN with SAE to deliver multi-building and multi-floor localization [6]. Different from the above existing solutions, this paper explores cross-device fingerprints at the signal level and the feature level. In addition, a capsule neural network and an edge-IoT framework are proposed to learn high-level spatial features from RSS fingerprints among heterogeneous IoT devices to achieve robust and real-time indoor localization.

## 5.3 The Architecture of Edge-IoT Framework for Heterogeneous IoT Devices

The system architecture of EdgeLoc is constructed for real-time indoor localization with heterogeneous IoT devices is shown in Fig. 5.1. EdgeLoc consists of three components: Wi-Fi APs, edge devices, and edge server. The details of the above

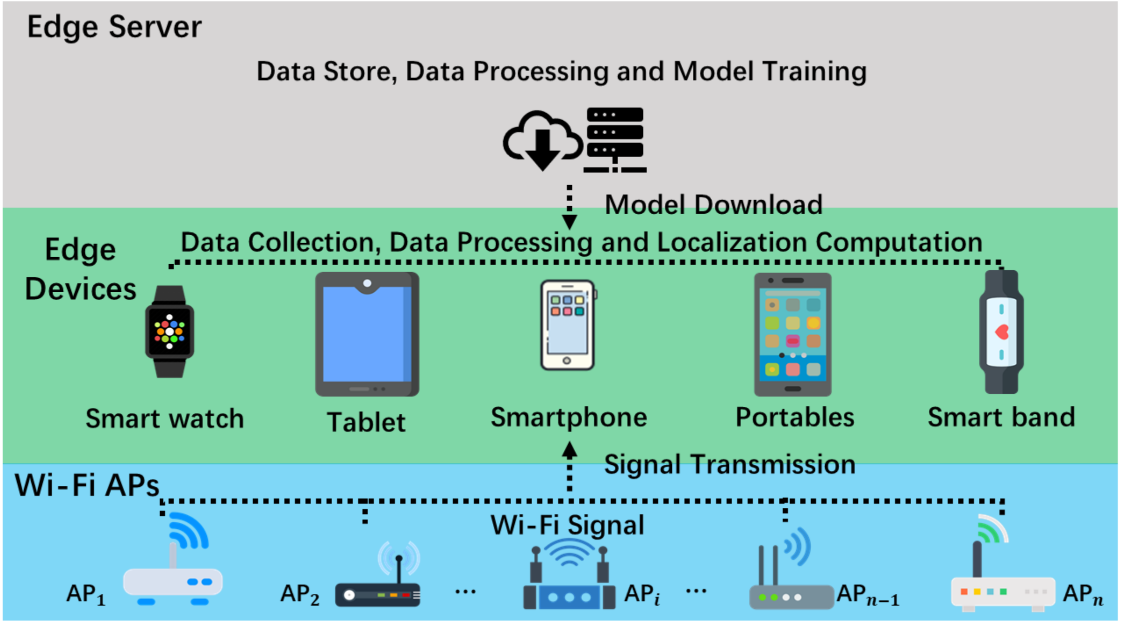


**Figure 5.1 .** The edge-IoT framework for real-time indoor localization towards heterogeneous IoT devices.

components are introduced as follows.

**Edge Server:** The edge server consists of a local database and a control center for edge-IoT indoor localization. The edge server is mainly responsible for data storage, data processing, and model training. Particularly, since the training process of the deep learning model is complicated and costly for edge devices, the CapsNet model is trained on the edge server. When the local database is updated, the parameter set is fed into the training model for re-training. For an edge device requesting localization service, the edge server will download the well-trained model parameters to the edge device. After that, each edge device can have its own model for light-weight localization.

**Edge Devices:** Heterogeneous IoT devices are ubiquitous with Wi-Fi connections in various indoor scenarios, and can perform edge computing tasks (*e.g.*, indoor localization) for mobile users. These devices, including smartphones, watches, tablets, and bands, generally have limited resources for computation and storage. Hence, edge IoT devices can only execute light-weight tasks for indoor localization, such as data collection, data processing, and location computation. It is also not feasible to perform model training on these edge devices [82]. As illustrated in Fig. 5.1, the edge devices collect Wi-Fi RSS data and leverage the optimized parameters of the deep learning model from the edge server for localization computation. In the experiments, a Raspberry Pi and an Android smartphone are used to test and verify these functions including light-weight data processing and computation for indoor localization. Note that the RSS fingerprints in the database are collected from heterogeneous IoT devices.

**Wi-Fi APs:** Wireless APs are the key components in EdgeLoc framework for indoor localization. Generally, Wi-Fi APs broadcast beacon frames to advertise their presence in the network (typically 100 ms per transmission). Upon scanning the channels to receive beacon information from surrounding Wi-Fi APs, mobile devices further calculate RSS from each AP [81]. RSS fingerprinting leverages RSS values (*e.g.*, a vector containing a series of RSS data) from multiple Wi-Fi APs as a unique fingerprint of the current location (*e.g.*, reference point). With RSS fingerprints from different locations stored in a local database, the location of the edge device (user reference point) can be estimated by finding the best match of its RSS measurement vector and the fingerprints of the anchor reference point [86].

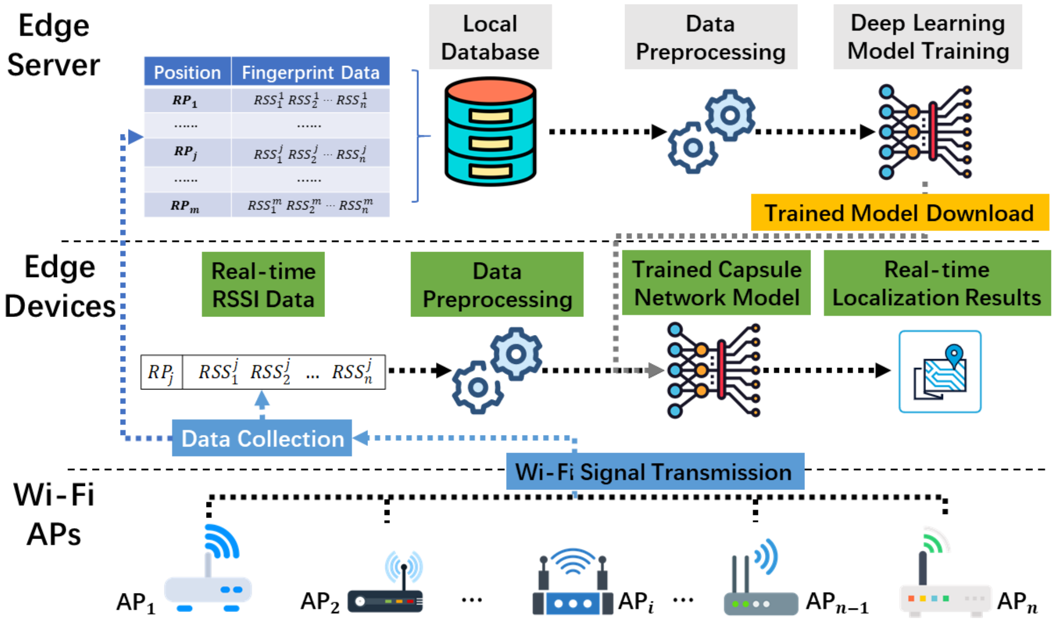

Edge Server
Position
Fingerprint Data
$RP_1$
$RSS_1^1$ $RSS_2^1$ ⋯ $RSS_n^1$
……
……
$RP_j$
$RSS_1^j$ $RSS_2^j$ ⋯ $RSS_n^j$
……
……
$RP_m$
$RSS_1^m$ $RSS_2^m$ ⋯ $RSS_n^m$
Local Database
Data Preprocessing
Deep Learning Model Training
Trained Model Download
Edge Devices
Real-time RSSI Data
Data Preprocessing
Trained Capsule Network Model
Real-time Localization Results
$RP_j$ $RSS_1^j$ $RSS_2^j$ … $RSS_n^j$
Data Collection
Wi-Fi Signal Transmission
Wi-Fi APs
$AP_1$
$AP_2$
⋯
$AP_i$
⋯
$AP_{n-1}$
$AP_n$

**Figure 5.2 .** The dataflow of EdgeLoc from the perspective of an edge-IoT platform.

## 5.4 The System Dataflow

In this section, the dataflow of the EdgeLoc system is described. As shown in Fig. 5.2. the overall dataflow involves two phases, *i.e.*, computation on the edge server and computation at edge devices.

For computation on the edge server, RSS fingerprint dataset is stored in a local database. The local database has the full historical RSS fingerprint data collected by the heterogeneous IoT devices. The raw data samples are then processed by the preprocessing module to construct the feature matrix. After that, the edge server starts training the deep learning model (CapsNet) with the labeled data to optimize model parameters for fingerprint-based indoor localization. For computation at edge devices, the RSS fingerprints are first collected by edge devices and are sent to the data preprocessing module. During the localization step, an edge device downloads the optimized model parameters from the edge server and employs the well-trained CapsNet model to perform real-time indoor localization.

The critical idea of system dataflow is to convert the RSS vector into the image-like data. The processed values of the vector correspond to the gray-scale in the image, which can directly reveal the influence of the indoor environment. Fig. 5.3 shows gray-scale images of normalized RSS vectors from the UJIIndoorLoc dataset [87], where fingerprints are collected from 25 heterogeneous Android devices. These devices are utilized to collect RSS data in different place at different time. Here, the dimensions of gray-scale images are set as the closest square to the number of selected APs for indoor localization [14]. For instance, the original RSS vector in Fig. 5.3 consists of 40 APs, so that the gray-scale image has 49 (closest square to 40) pixels. Correspondingly, the dimensions of the image are set to $7 \times 7$, where 9 pixels with zero intensity are padded at the end of the original RSS vector to increase the size from 40 to 49. By following the above procedure, RSS fingerprinting images in Fig. 5.3 are converted from the normalized vectors at the same location, collected by Android devices of numbers 0, 9, 13, and 15. These images carry the most relevant features that can be learned by the proposed CapsNet model for indoor localization, such as the hierarchical feature of RSS data collected by various devices. However, some features in image-like RSS data still contain fluctuations caused by dynamic environment or multipath effects. The data preprocessing procedures as described in Section 4.4.1 are proposed to further improve the representativeness of input data.

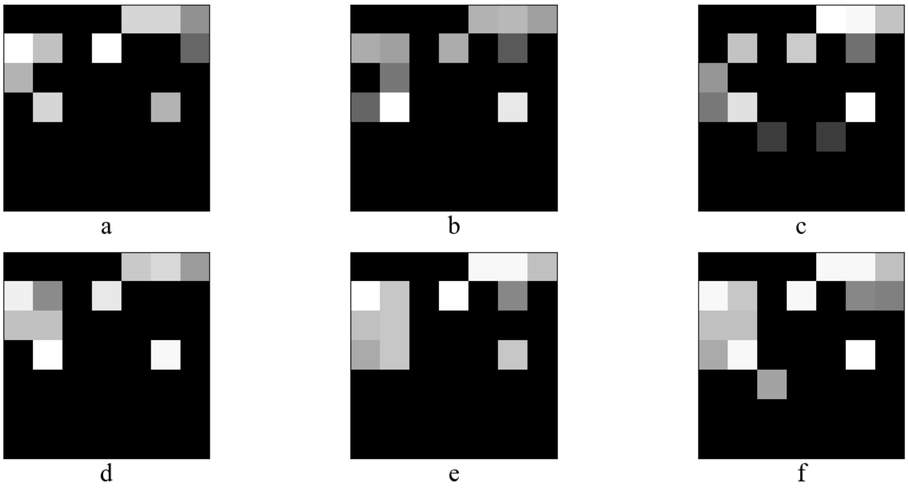


**Figure 5.3 .** The gray-scale image of normalized RSS vectors of 40 APs, sampled at the same location by different Android devices. Images (a) and (c): data collected device 13; Images (e) and (f): data collected device 15; image (b) and (d): data collected by devices 0 and 9, respectively.

### 5.4.1 Dataflow in Model Training on the Edge Server

The flowchart of the training phase on the edge server is shown in Fig. 5.4. This subsection introduces the dataflow and parameters of the CapsNet model from the input to the output as follows:

- The input layer takes the feature matrix $R$ into the model, which is down-sampled to the size of $n \times n$.
- The second layer is a Conv layer, where the size of the convolution kernel is $3 \times 3$ and the stride is 1. Here, the number of filters in the CNN layer is to be learned.
- The third layer is a PC layer. Similarly, the kernel for the convolutional

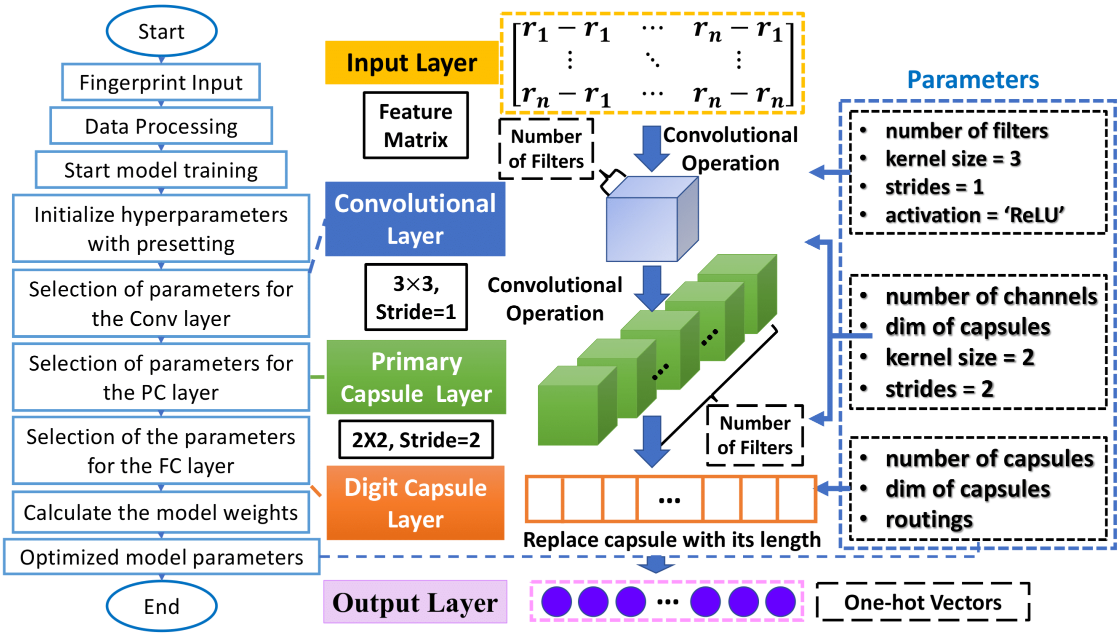


**Figure 5.4 .** The flowchart of the proposed methodology and the dataflow/parameters in the CapsNet model.

operation in this layer is 2 × 2, but the stride is 2. The number of channels (*i.e.*, filters) and the dimension of the capsule in this layer are to be learned.

- The fourth layer is a FC layer, where the number of capsules is equal to the number of grids in the experimental area (as illustrated in Section 4.4.1). The dimension of a capsule here is the same as that in the PC layer.

- The last layer is the output layer, which replaces each capsule with its length to match the label's shape. The dimension of the output is the same as in the FC layer.

### 5.4.2 RSS Fingerprinting Data

The method utilizes RSS data collected from Wi-Fi APs in an indoor environment to create an RSS fingerprint database. Assuming that there are $n$ APs and $k$ reference points across the indoor space, at each point, $r$ RSS samples are collected. Each RSS sample is labeled by the two-dimension location information (*i.e.*, row and column) as the ground-truth. In this way, the unrolled fingerprint database can be considered a huge matrix with $m \times n$ vectors ($m = r * k$).

## 5.5 Experimental Studies

In experimental studies, a prototype system of EdgeLoc is deployed in Building 1 at Beijing University of Posts and Telecommunications. The floor plan and deployed Wi-Fi APs are illustrated in Fig. 3.5 as shown in Section 3.4.

### 5.5.1 Experimental Setup

**System Setup.** As illustrated in Fig. 5.5, a Dell Latitude 5480 laptop is employed as edge server, with a Raspberry Pi 3 and a Redmi smartphone working as edge IoT devices. The edge server has a 4-thread Intel i7-7600U CPU of 2.9 GHz and 16GB RAM. For edge IoT devices, the Raspberry Pi has a 64-bit quad-core ARMv8 CPU and the Redmi smartphone is equipped with a 64-bit Qualcomm Kryo 470 CPU. The localization model of EdgeLoc is implemented on Keras framework of TensorFlow using Python 3.6.

**Data Collection.** In a typical indoor environment, fluctuations of measured RSS signals always reduce the accuracy and stability of indoor localization. BUPT

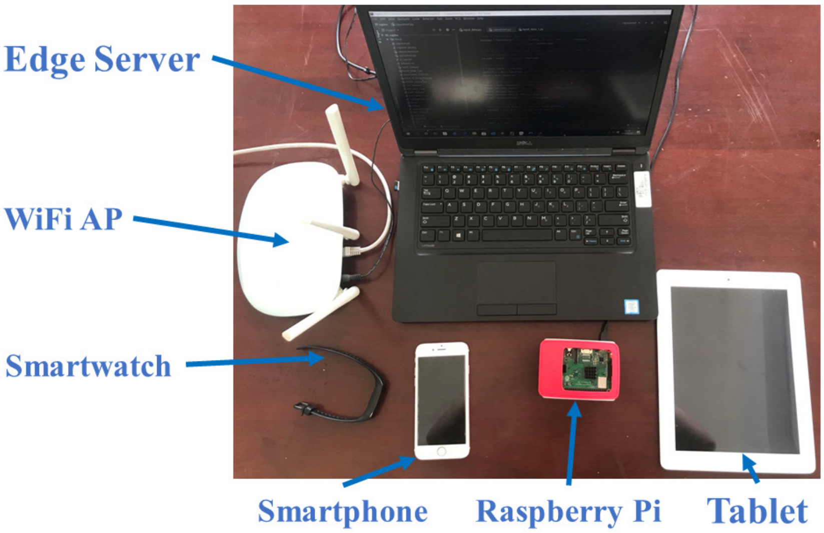


**Figure 5.5 .** The system implementation of EdgeLoc.

database is collected as shown in Section 4.5.2. However, UJIIndoorLoc RSS database are collected from edge devices rather than laptop because device heterogeneity problem is considered.

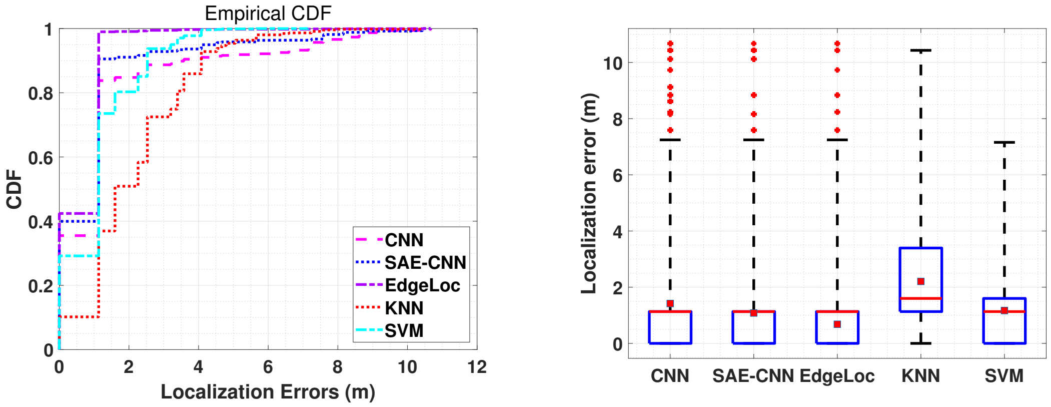
**Baseline Methods.** To fully evaluate the performance of EdgeLoc, representative baseline methods of indoor localization are adopted for comparison as shown in Section 4.5.3.

**Experiment evaluation indicators.** The experiment evaluation indicators of this chapter include localization error, average localization error, localization accuracy, mean positioning(prediction) time. Localization error is the Euclidean distance between estimated coordinates and actual ones of the same location. Average localization error is the mean value of localization error of all testing data. Localization accuracy is the probability that estimated location classification of testing data is correct. Positioning time is the time that one testing data is processed by all online modules. Mean positioning time is the mean time of all testing data.

**Parameter Settings.** The dataflow of the CapsNet model is presented in Fig. 5.4. The input and the output of each layer are tensors that share the same data format. The main parameters of each layer are listed in the Parameters block. In specific, for the Conv1 layer, the convolutional kernel size is set as 3, the convolutional stride is set as 1, and the activation function is 'ReLU'. The impact of different numbers of filters *(n_filters)* is evaluated in experiments. For the PC layer, the convolutional kernel size is set as 2, the convolutional stride is set as 2, and the activation function is 'Squash'. Similarly, the impact of the numbers of channels *(n_channels)* and the dimension of the capsule *(dim_capsule)* are studied. For the FC layer, the number of routing iterations is set to 3, the number of capsules equals the number of grids in the localization area and the impact of *dim_capsule* are evaluated. Note that the values of *dim_capsule* for both PC and FC layers are the same.

### 5.5.2 Performance on BUPT Database at the IoT Edge Devices

**Overall Localization Performance.** The performance of EdgeLoc in comparison with other baseline methods is shown in Fig. 5.6. Overall, EdgeLoc outperforms all the baseline methods, where 99% of testing results are within errors of lower than 2 $m$, and over 40% of testing samples are with errors of around 1 $m$. In contrast, as shown in the Cumulative Distribution Function (CDF) distribution of Fig. 5.6a, FS-$k$NN [54] has the most unsatisfactory performance with more than 60% local-



**(a)** The CDF distribution of localization errors **(b)** The box-plot distribution of localization errors

**Figure 5.6 .** Comparison of localization accuracy by EdgeLoc with 8 channels and 16 channels.

ization errors from 2 $m$ up to 8 $m$. For SVM [19], nearly 20% of testing results are with errors larger than 2 $m$ and 5% of testing results are with errors larger than 3 $m$. Moreover, CNN [65] extracts high-level features to enhance localization accuracy, resulting in less than 2 $m$ in 80% of testing data. SAE-CNN [5] further improves the performance by encoding raw RSS fingerprints into high-level features for CNN. Fig. 5.6b further depicts a box-plot for the localization results. Compared with CNN and SAE-CNN, the proposed EdgeLoc has the smallest IQR (*i.e.*, the distance between first quartile and third quartile) and the lowest median localization error of 0.68 $m$. In contrast, KNN-based method shows the worst performance with the largest IQR and the highest median error. As the existing baseline methods, CNN and SAE-CNN models have similar performance to EdgeLoc overall except for higher median errors. The above results show that by preserving valuable spatial

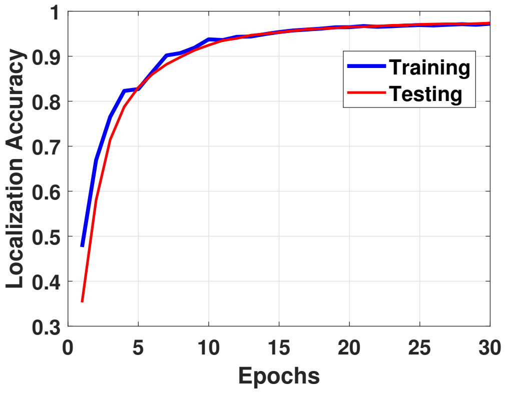
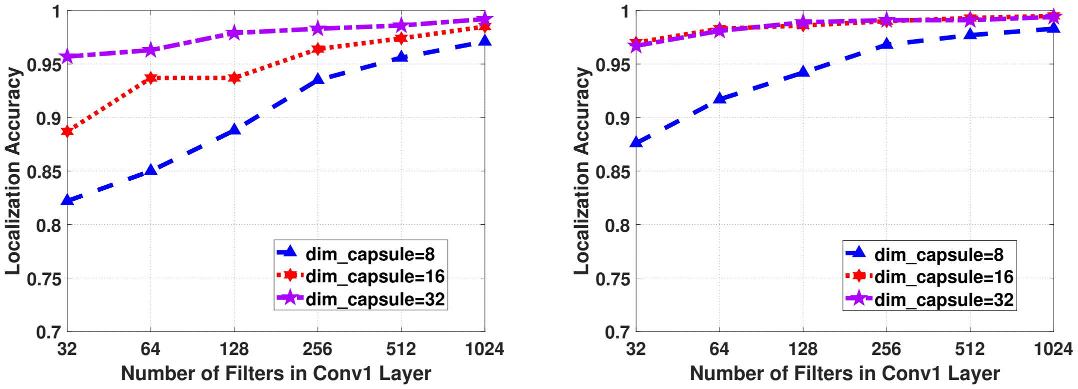


**Figure 5.7 .** Localization accuracy with training dataset and testing dataset.

RSS fingerprinting information in between-layer neurons, EdgeLoc is superior to conventional machine learning models and existing deep learning models.

To achieve the best trade-off of EdgeLoc's model parameters, localization accuracy and positioning time are evaluated, respectively. Before tuning parameters of the CapsNet model, Fig. 5.7 visualizes the localization accuracy of EdgeLoc in the training session and testing session. It shows that EdgeLoc can achieve over 90% localization accuracy after 8 epochs for both training data and testing data.

**Parameters' impact on the localization accuracy:** In the PC layer, the number of channels (*i.e.*, *n_channels*) is set as 8 and 16 for evaluation, respectively. The dimension of the capsules (*i.e.*, *dim_capsule*) ranges from 8 to 16 and 32. Moreover, a grid search is then conducted for the number of filters (*i.e.*, $n\_filters$) in the Conv1 layer, over the parameter set of {32, 64, 128, 256, 512, 1024}. Fig. 5.8 shows the evaluation results on the parameter's impact on localization accuracy. First, the overall localization accuracy improves steadily with a larger *n_filter* in the Conv1 layer. Meanwhile, as the *n_channels* increases from 8 to 16, the localization



**(a)** PC layer with 8 channels

**(b)** PC layer with 16 channels

**Figure 5.8 .** Comparison of localization accuracy by EdgeLoc with 8 channels and 16 channels.

accuracy is not significantly enhanced. Second, the *dim_capsule* directly impacts the indoor localization accuracy, as EdgeLoc shows an improvement of up to 10% in Fig. 5.8a and Fig. 5.8b. Meanwhile, with the increasing value of *n_filters* in the Conv1 layer, the improvement of localization accuracy shrinks to nearly 5%. The above evaluations give an insight into the contributions of different components in EdgeLoc, where the best setting of EdgeLoc is with 64 filters in the Conv1 layer and 8 capsules, with a dimension of 16 in each capsule in the settings.

Fig. 5.9 shows the localization accuracy of EdgeLoc with different training samples and different batch sizes during the training process, where ”batch sizes” means the size of each batch of data. Here, $\beta$ denotes the size of the training samples in proportion to the overall RSS fingerprinting dataset. As revealed by the experimental results in Fig. 5.9, with a larger size of the training samples, the overall localization accuracy significantly improves (*e.g.*, up to 7.2% when batch size is 50). Moreover,

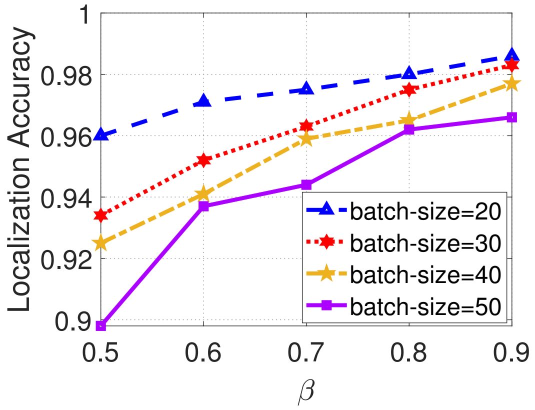
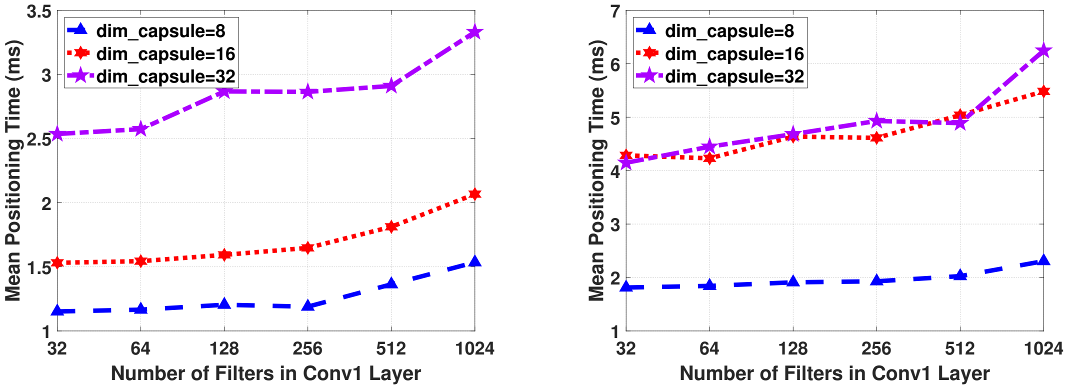


**Figure 5.9 .** Localization accuracy of EdgeLoc by varying batch size and training samples.

taking $\beta = 0.5$ as an example, the performance of EdgeLoc decreases from 0.96 to 0.9 with batch sizes of 20 and 50 accordingly.

**Parameters' impact on the mean positioning time:** Similar to the evaluation of localization accuracy, two cases are tested, where the *n_channels* of the PC layer are 8 and 16, and with other parameters remaining the same. As shown in Fig. 5.10a and Fig. 5.10b, the positioning time has positive correlations to the *n_channels* in the PC layer, larger *dim_capsule*, and larger *n_filters* in the Conv1 layer.

Fig. 5.11 depicts the mean positioning time of the EdgeLoc, where CapsNet model is with 8 channels, 8-dimension capsules and varying numbers of Conv1 filters. Here, the positioning time is strongly correlated to the number of filters in the Conv1 layer. The reason is that the Conv1 layer has a more powerful representation learning capability with larger filters, thereby improving the response time in calculating the matched reference point from the fingerprinting database [5].



**(a)** PC layer with 8 channels

**(b)** PC layer with 16 channels

**Figure 5.10 .** Comparison of positioning time by EdgeLoc with 8 channels and 16 channels.

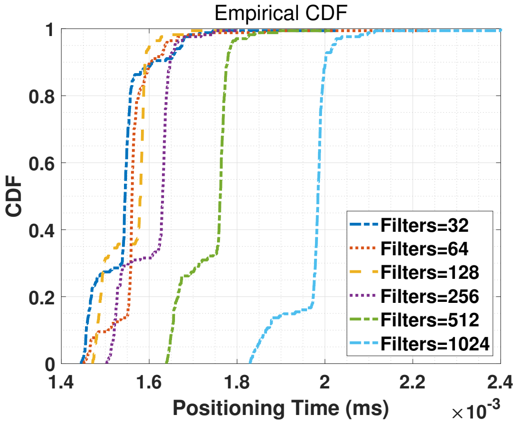


**Figure 5.11 .** The CDF of positioning time in all testing dataset.

In addition, Table 5.1 shows the mean positioning time *vs.* batch size, where EdgeLoc is with 64 filters in the Conv1 layer, 8 capsules with 16-dimension at the edge. When the batch size becomes more extensive, the mean positioning time is reduced from 2.05 $ms$ to 1.60 $ms$.

To this end, by jointly considering the performance of localization accuracy and the mean positioning time, the EdgeLoc model with 1024 filters in the Conv1 layer and 8 capsules with 16-dimension can achieve the best trade-off between localization

Table 5.1 : The mean positioning time of EdgeLoc *vs.* batch size.

| Batch Size | 20 | 30 | 40 | 50 |
|---|---|---|---|---|
| Mean Positioning Time ($ms$) | 2.05 | 1.85 | 1.72 | **1.60** |

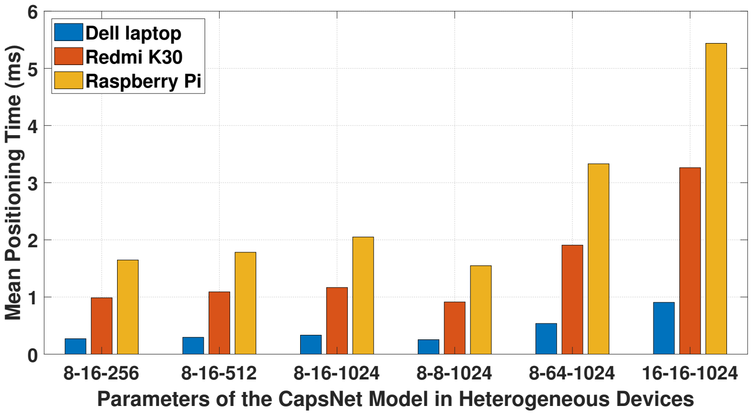


**Figure 5.12 .** The mean positioning time by different devices with different model parameters on BUPT dataset.

accuracy and positioning time on the edge server, with the localization accuracy of 98.5% and the average positioning time of 2.05 $ms$.

**Impact of Device Heterogeneity on Localization Time:** The last experiment compares the localization time of two heterogeneous IoT devices with the edge server using BUPT dataset. In Fig. 5.12, a Raspberry Pi and a Redmi K30 smartphone are employed as two edge devices to perform indoor localization tasks. Note that for the same parameter setting on the above two types of devices, EdgeLoc has the same localization accuracy and demonstrates its scalability towards heterogeneous IoT devices. For example, combination of '8-16-1024' means that parameter setting in CapsNet model is: $n\_channels = 8, dim\_capsule = 16$, and $n\_filters = 1024$. Due to different computation capabilities, edge devices show higher positioning time than the edge server (*e.g.*, a Dell laptop). Similarly, as the Redmi K30 smartphone has better computation power than the Raspberry Pi, it only needs half of the computation time of the Raspberry Pi's. Nevertheless, the average computation time for all devices is only within a few milliseconds. It further validates the effectiveness of the proposed edge-IoT framework for indoor localization with heterogeneous IoT devices.

### 5.5.3 Extensive Experiments on the UJIIndoorLoc Dataset at the IoT Edge Devices

**Scalability of the CapsNet on the UJIIndoorLoc dataset:** To further study the scalability of EdgeLoc, UJIIndoorLoc dataset is applied for performance evaluation [87]. UJIIndoorLoc dataset covers three different buildings (with ID 0, 1, and 2) of more than 110,000 $m^2$ indoor areas, with 19,937 training samples and 1111 test samples of RSS fingerprints. These samples are collected by heterogeneous IoT devices (25 Android devices) and contain random variances. In particular, the Building 0 from UJIIndoorLoc dataset is chosen to evaluate EdgeLoc and the top-40 APs (out of a total 520 APs) are selected to characterize RSS fingerprints by ranking all APs' frequency of occurrence in descending order. The localization performance of all baseline methods is presented in Table 5.2, where term ”level” is the level number of the building. EdgeLoc achieves the best localization performance and outperforms the existing SAE-CNN [5] by up to 14.4% at level 3.

Table 5.2 : The average localization errors ($m$) of various models on the UJIIndoorLoc dataset.

| Level / Models | 0 | 1 | 2 | 3 | all |
|---|---|---|---|---|---|
| **EdgeLoc** | **8.28** | **7.36** | **8.32** | **7.75** | **7.93** |
| KNN [54] | 8.43 | 7.50 | 8.61 | 7.86 | 8.10 |
| SVM [19] | 8.85 | 7.68 | 9.49 | 8.14 | 8.54 |
| CNN [65] | 8.59 | 8.15 | 8.85 | 7.79 | 8.35 |
| SAE-CNN [5] | 8.43 | 8.27 | 9.12 | 9.05 | 8.72 |

**Impact of the number of Wi-Fi APs on the localization accuracy for the UJIIndoorLoc dataset:** Table 5.3 shows the average localization errors of EdgeLoc by adopting different numbers of APs in the UJIIndoorLoc dataset for indoor localization. The 520 APs are ranked in descending order by their frequency of occurrence and this chapter selects the top 20, 30, 40, and 50 APs to generate fingerprint data, respectively. As revealed in Table 5.3, with more APs, EdgeLoc shows higher accuracy in performing localization. Besides, the performance of EdgeLoc converges with 40 APs, which demonstrates that the RSS fingerprints from top-40 APs are already sufficient for EdgeLoc to extract representative features in localization.

Table 5.3 : The average localization error ($m$) of EdgeLoc based on different numbers of Wi-Fi APs in Building 0.

| Level<br>APs | 0 | 1 | 2 | 3 | all |
|---|---|---|---|---|---|
| 50 APs | 8.63 | 7.53 | **8.26** | **7.70** | 8.03 |
| 40 APs | **8.28** | **7.36** | 8.32 | 7.75 | **7.93** |
| 30 APs | 9.52 | 8.39 | 8.58 | 9.32 | 8.95 |
| 20 APs | 10.61 | 9.30 | 8.98 | 9.36 | 9.56 |

## 5.6 Summary

In this Chapter, a novel, robust and real-time indoor localization solution towards heterogeneous IoT devices called EdgeLoc is proposed. Furthermore, an edge-IoT framework is presented to enable heterogeneous IoT devices to share optimized model parameters from the edge server for real-time localization. The solution further addresses the superimposed and typical challenges through signal-level and feature-level data processing, respectively. EdgeLoc is set up in a real-world experimental field and extensive experiments are conducted. Evaluation results show that EdgeLoc can achieve up to 98.5% accuracy for indoor localization at an average positioning time of only 2.05 $ms$.

# Chapter 6

# *SE-Loc*: Security-Enhanced Indoor Localization with Semi-Supervised Deep Learning

Wireless indoor localization has become one of the key technologies for industrial ILBSs. Given ubiquitous deployment of Wi-Fi networks, Wi-Fi fingerprinting of RSS has been widely adopted in indoor localization. Meanwhile, existing RSS fingerprint-based methods lack security awareness and are vulnerable to malicious attacks. When security vulnerabilities are concerned, mobile users may confront indoor localization mismatches, faults and even localization system failures. This chapter proposes *SE-Loc*, a semi-supervised learning-based technique to enhance security and resiliency of fingerprint-based localization. The architecture of *SE-Loc* consists of two parts: (1) a correlation-based AP selection for processing RSS fingerprints and fingerprint-image generation, and (2) a deep learning model based on a denoising autoencoder and convolutional neural network for robust feature learning and location matching. Extensive experiments show that under potential attacks on Wi-Fi networks, *SE-Loc* demonstrates superior performance on indoor localization over state-of-the-art methods. With up to 100 malicious attacks at APs via UJIIndoorLoc edge server, *SE-Loc* can still achieve the lowest error fluctuation of 1.7 m and the lowest average localization error of 8.9 m.

## 6.1 Introduction

With the rising of the IoT industry, indoor localization has become a key enabler to many ILBS, such as indoor navigation [67] and motion detection [84]. To deliver pervasive wireless localization, Wi-Fi networks have become one of the commonly utilized solutions due to wide availability with IoT devices. Moreover, Wi-Fi RSS fingerprinting has received considerable interests for achieving non-line-of-sight industrial indoor localization. Intuitively, an indoor location can be identified based on a unique RSS vector collected from different Wi-Fi APs. In general, Wi-Fi fingerprint-based localization consists of two phases [42]: (1) an offline phase to collect RSS data at the locations of interest and construct the fingerprint dataset (*i.e.*, radio maps) for model training; (2) an online phase to locate users in real-time by matching their fingerprint observations with radio maps in the dataset. Besides, Wi-Fi fingerprint-based indoor localization has been integrated into commercial mobile applications by various industrial counterparts, including Google Maps, Apple Maps and Bing Maps [88].

To improve localization accuracy, many techniques that are based on RSS fingerprints have been developed to accurately match user-observed fingerprints with radio maps [42]. More recently, deep learning models are introduced to extract incremental representations from the fingerprinting inputs. The typical deep learning-based indoor localization models include WiDeep [89], CNNLoc [5], EdgeLoc [3], and iToLoc [90]. The above models are built to achieve efficient and effective localization for various industrial scenarios through jointly learning features from complex

RSS fingerprints in a layer-by-layer manner. Despite that high-level and useful fingerprinting features are expected from the learning process of indoor localization, *security* is considered as an underline premise for reliability of industrial indoor localization [91].

The generic security issue of industrial indoor localization refers to external threats and AP attacks that bring severe RSS fluctuations and further result in abrupt changes in fingerprints, which has been investigated [14]. As suggested by [92], RSS-based indoor localization techniques are particularly vulnerable to security threats, such as AP jamming, AP spoofing, Wi-Fi signal interference and malicious AP impersonation [91]. When confronting the above security threats, RSS fingerprints of legitimate APs can be severely changed, leading models to generate localization results with large errors and even wrong locations. Therefore, it remains daunting to deal with security vulnerabilities of RSS-based indoor localization with the following challenges.

**Challenge 1**: *Taming unpredicted RSS fluctuations under random attacks on Wi-Fi APs*: To bypass security checks and avoid detections, most malicious AP attacks would be random and almost imperceptible for both online and offline phases of indoor localization [93]. Therefore, it is challenging to tame unpredicted RSS fluctuations and eliminate abrupt changes in fingerprints caused by malicious attacks.

**Challenge 2**: *Achieving security enhancement without compromising effectiveness and efficiency of indoor localization*: ILBS demands high accuracy and time efficiency in indoor localization. However, integrating any attack detection module into localization framework can significantly increase computing complexity of indoor localization process [94]. Besides, the possible misjudgments of attack detection module may label legitimate APs identified as malicious ones, thus degrade localization accuracy. Hence, it remains challenging to enhance security without compromising the performance of indoor localization.

Existing works only partially solve security challenges from perspectives of data extrapolation (S-CNNLOC [14]), feature selection (RMBMFL [92]) and privacy preservation (PILOT [95]). This chapter proposes *SE-Loc*, a security-enhanced framework with semi-supervised deep learning, aiming to deal with the above challenges simultaneously. *SE-Loc* leverages signal-level and feature-level enhancements to secure indoor localization against potential AP attacks, respectively. To tame abnormal RSS fluctuations and abrupt changes in fingerprints, *SE-Loc* employs the Pearson Correlation Coefficient (PCC) method in AP selection and fingerprint image generation. In addition, *SE-Loc* combines a DAE with CNNs for effective feature extraction and representation learning on RSS fingerprints. First, DAE model component denoises sharp changes (*e.g.*, abnormal increase or decrease) in RSS values and extracts robust features to reduce the risk of being influenced by malicious attacks. Second, the CNNs model is responsible for accurately and efficiently matching the user-observed fingerprints with the radio map. Overall, *SE-Loc* achieves reliability and enhances security of indoor localization process.

The main contributions of this chapter are summarized as follows:

- To the best of the knowledge, it is the first work to enhance security of indoor localization with a deep learning model that consists of DAE and CNN.

In particular, this chapter addresses security and reliability issues of Wi-Fi fingerprint-based indoor localization simultaneously.

- At the signal level, a rigorous and efficient computing process for PCC-based AP selection is conducted to eliminate APs that are attacked and contaminated. At the feature level, this chapter proposes a deep learning model that combines DAE with CNN for effective feature extraction and efficient representation learning on RSS fingerprints.
- Extensive experiments show that when confronting up to 100 malicious attacking APs in the UJIIndoorLoc dataset, *SE-Loc* performs the best among all state-of-the-art baselines, with the lowest error fluctuation and the highest average localization accuracy.

The rest of this chapter is organized as follows: Section 6.2 reviews up-to-date related work. Section 6.3 introduces the preliminaries of fingerprinting-based indoor localization under malicious AP attacks. Section 6.4 proposes the architecture of the *SE-Loc*. Section 6.5 introduces detailed implementation and data processing procedures. Section 6.6 presents extensive experimental results and the comprehensive analysis of the implemented *SE-Loc*. Section 6.7 concludes the chapter.

## 6.2 Related Work

**Feature extraction for Wi-Fi RSS fingerprints.** Wi-Fi RSS fingerprint-based indoor localization techniques are resistant to multi-path effects and radio frequency shadowing effects [96]. Meanwhile, signal variations and device heterogeneity remain the primary obstacles to localization accuracy. This attracts subsequent research on various methods for feature enhancements on Wi-Fi RSS fingerprints, including feature scaling [54], signal calibration [97] and multivariate linear regression [70]. For instance, [98] proposed Ellipsoid features by employing tuples of pairs of RSS values to eliminate device dependence and compensate performance degradation. More recently, spatial and temporal features have been leveraged for indoor localization and relevant extractors have been developed. Li *et al.* [99] exploited spatial awareness in Wi-Fi fingerprints and used spatial features of RSS from nearby locations to mitigate spatial ambiguity and temporal instability of fingerprints in indoor localization. Different from the above, *SE-Loc* utilizes correlation-based AP selection, fingerprint-image generation and mutual difference-based feature matrix for reliable feature extraction.

**Deep learning models for accurate indoor localization.** To achieve accurate indoor localization, various deep learning models have been applied to effectively learn RSS fingerprinting features [100]. [5] proposed CNNLoc, a convolutional neural network-based indoor localization system for multi-building and multi-floor localization. Based on CNNs, many variant models have been further developed to solve different issues in indoor localization, including the WiDeep [89], StoryTeller [101], CapsLoc [2], *etc.* To enable automatic indoor radio map construction and model adaption, adversarial learning methods have been further adopted [102]. [90] designed a domain adversarial learning framework to solve deterioration problems, which can automatically learn indoor localization model through the co-training process. [103] proposed a feature-metric point cloud registration framework with encoder and multi-task branches, so as to minimize projection errors and enable spatial localization in 3D scenes with point cloud registration. [104] presented a human activity recognition scheme using the matching network with enhanced channel state information to facilitate one-shot learning. Motivated by the above advances of deep learning, *SE-Loc* combines a denoising autoencoder and convolutional neural networks to learn attack-resistant representations for indoor location matching.

**Security-oriented solutions for indoor localization.** Apart from the accuracy and robustness, security issue becomes another major challenge to the reliability of indoor localization, as most existing methods are quite vulnerable to AP-oriented attacks [14]. Yuan [91] identified two practical RSS attacks on fingerprinting-based indoor localization and further designed a fingerprint-matching mechanism with a novel truncated distance metric. Moreover, Saideep *et al.* [14] modified the existing technique CNNLoc [5] with an offline fingerprint dataset extrapolation method to improve the resiliency of indoor localization. Li *et al.* [105] proposed an optimized multi-voting mechanism to defend indoor localization against physical-layer signal strength attacks. Different from the above, *SE-Loc* leverages signal-level and feature-level enhancements for secure indoor localization against potential attacks, which is more robust to overcome random AP attacks and is more effective to maintain the efficiency of localization.

## 6.3 Preliminaries of fingerprinting-based indoor localization under malicious AP attacks

Traditional Wi-Fi RSS fingerprinting system consists of an indoor Wi-Fi network with $N$ APs. An RSS vector sent from these APs is represented as a vector $RSS = [RSS_1, \ldots RSS_i, \ldots RSS_N]$, where $RSS_i$ represents the measured RSS from the $i$th AP. These collected data are utilized to estimate location by RSS fingerprinting-based indoor localization methods. A typical fingerprinting localization process can be divided into two phases: In the offline phase, for each node in a set of $K$ nodes, collected RSS data has the form of $[RSS^k, l^k]$, where $RSS^k$ represents the RSS vector collected at the $k$th node ($l^k$). Collected data is stored in an RSS dataset as training data. In the online phase, a device measures fingerprinting vector $\widetilde{RSS}$ and uploads it to the server of indoor localization system. By matching the $\widetilde{RSS}$ with the established radio map in the dataset, the server sends back an estimated location $\tilde{l}$ to a mobile device.

Conventional Wi-Fi RSS fingerprinting-based localization methods are unreliable due to false fingerprint problem in smart buildings. The false fingerprint problem is mainly caused by environmental factors or even security threats. Taking a local zone from UJIIndoorLoc dataset as an example, there are passive interference (environmental factors) and active interference (security threats), as shown in Fig. 6.1.

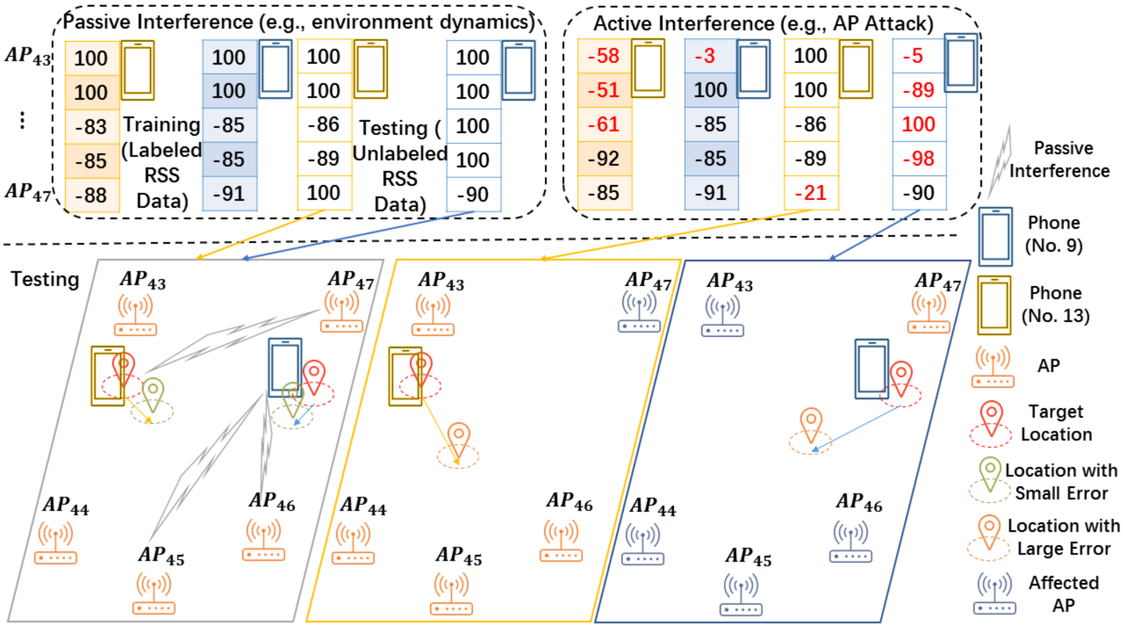


**Figure 6.1 .** An illustration of passive (environmental) interference and active (AP attacks) interference cases for RSS fingerprinting-based indoor localization.

### 6.3.1 RSS Path Loss Model with Passive Interference

Once the noise caused by multi-path interference and device heterogeneity (passive interference) in complex indoor scenario as shown in Fig. 6.1 is defined, wireless signal propagation model popularly called the path loss model towards heterogeneous IoT devices can aptly reflect the shadow effect statistically as shown in the equation below:

$$RSS_i[dB] = RSS_i(d_0) + 10\eta log(\frac{d_i}{d_0}) + n_{\sigma i} + n_s, \tag{6.1}$$

where $\eta$ is the path loss exponent, $d_0$ is the reference distance, and $RSS_i(d_0)$ is the reference path loss from the $i$th Wi-Fi AP, which can be calculated using the path loss formula. The parameter $n_i$ is a zero-mean Gaussian random variable (in dB) with standard deviation $\sigma_i$. And $n_s$ is the random variable (in dB) from heterogeneous IoT devices.

### 6.3.2 RSS Measurements with Malicious AP Attacks

Apart from the environmental interference, this chapter further addresses the vulnerabilities of RSS fingerprinting-based indoor localization by considering malicious AP attacks. As shown in Fig. 6.1, attackers could deploy attacking AP nodes to actively interfere RSS measurements of legitimate APs. This section summarizes vulnerabilities of RSS fingerprinting that can be exploited to attack indoor localization as follows.

- Man-in-the-Middle (MitM) Attacks. The MitM attackers set up fraudulent access points and configure them with the same SSID (Service Set Identifier), IP (Internet Protocol) and even MAC address as the legitimate ones [106]. The adversary APs can be adjusted to have significantly stronger signals to cover the original RSS and ultimately influence RSS fingerprints. When confronting MitM attacks, the offline phase of indoor localization will induce inevitable noises in the fingerprint dataset and the online phase will receive mismatched localization results.

- Distributed Denial of Service (DDoS). The DDoS attackers aim to disrupt the availability of system resources to legitimate users [107, 108]. For the indoor localization scenario, an adversary can perform DDoS attacks on physical layer or the data link layer, by jamming the radio frequencies or spoofing packets. For instance, an adversary jammer emits malicious Wi-Fi signals to occupy a wireless channel and produces signal interference with legitimate Wi-Fi APs on the same channel. Correspondingly, the localization system or mobile users will lose visibility of attacked APs and the localization accuracy will be compromised without causing any notice.

- Key-recovery attacks. The key-recovery attackers aim to associate with a Wi-Fi network by using a range of crack techniques [109]. The recovered key would provide the adversary with capabilities to exploit various vulnerabilities against indoor localization procedure, launching Evil Twin attacks and Address Resolution Protocol spoofing to make interference. In addition, mobile user's location will be leaked to the adversary, causing further privacy and even personal security issues.

- Dataset corruption. Based on the above attacking procedures, the malicious third party can have chances to corrupt fingerprint dataset, by changing RSS values associated with a large scale of legitimate APs [110]. Since the established radio maps contain critical information for indoor localization, it would significantly jeopardize the whole indoor localization system by destructing the reliability of the fingerprint dataset.

Given the above attack modes, the key impact of various AP attacks is to change RSS values observed by mobile users. These changes appear as abrupt, random, fluctuating and undesirable noises to indoor localization systems. As traditional machine learning methods can not handle these, their localization performance can degrade significantly when confronting malicious AP attacks. To maintain resistance of indoor localization with external attacks, this chapter adopts semi-supervised deep learning method to take random changes of both labeled and unlabeled RSS data into consideration, thus taming the performance degradation caused by potential AP attacks. By referring to the existing model in [14], this chapter introduces the RSS measurement function considering AP attacks as follows.

$$RSS_i^{Attack}[dB] = \begin{cases} I & \text{if there is an attack on the } i\text{th AP}, \\ RSS_i & \text{otherwise}, \end{cases} \tag{6.2}$$

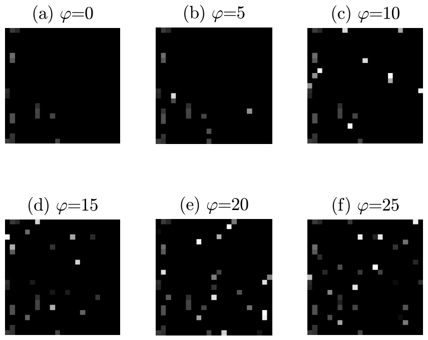
where $I \sim U\{-100, 0\}$ ($I$ ranges from -100 to 0 randomly.) denotes attack perturbation from the $i$th Wi-Fi AP, and it is not zero only when the $i$th Wi-Fi AP attack is malicious. $RSS_i$ is described in the Equation 6.1.

### 6.3.3 Normalization for RSS Measurements with Malicious AP Attacks

As a basic step to tame tainted RSS measurements under malicious attacks, this chapter normalizes the raw RSS data through the following data processing procedure.

$$r_i = \begin{cases} 0 & RSS_i^{Attack} = 0, \\ 0.1 * (I - \min) & \text{if there is an attack on the } i\text{th AP}, \\ 0.1 * (RSS_i - \min) & \text{otherwise}, \end{cases} \tag{6.3}$$

where $RSS_i$ is the raw RSS value from AP $i$, $r_i$ is the normalized RSS value of AP $i$, $I \sim U\{-100, 0\}$ is the perturbation factor for attacked AP and min is the lowest RSS value among all raw RSS values.


(a) $\varphi$=0 (b) $\varphi$=5 (c) $\varphi$=10

(d) $\varphi$=15 (e) $\varphi$=20 (f) $\varphi$=25

**Figure 6.2 .** The fingerprint image of the normalized RSS data with varying $\varphi$ at level 2 building 0, UJIIndoorLoc dataset.

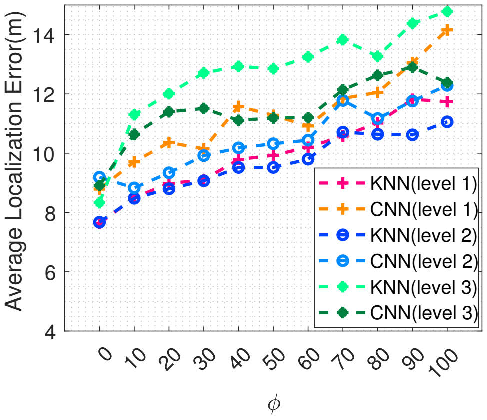


**Figure 6.3 .** The impact of the upper limit of AP attacks ($\phi$) against K-Nearest-Neighbor-based and CNN-based indoor localization methods.

### 6.3.4 RSS Fingerprinting-based Indoor Localization under Malicious AP Attacks

As shown in Equation 6.2, $I$ is not related to the distance $d_i$ between the $k$th user location and the $i$th Wi-Fi AP, when the $i$th Wi-Fi AP attack is malicious. In other words, $I$ is interference factor for location estimation, which is not related to the location $l^k$. Specially, the interference values of unlabeled RSS data are different from that of labeled ones, which can increase the localization error of unlabeled data by traditional RSS fingerprinting-based methods.

Fig. 6.2 shows the gray image of the normalized RSS data (described in Section 6.4.3) collected at location $l_k$ with different numbers of malicious AP attacks ($\varphi = 0, 5, 10, 15, 20, 25$) on the UJIIndoorLoc dataset of level 2 building 0. Fig. 6.3

shows the average localization error of CNN method utilizing normalized RSS data of the UJIIndoorLoc dataset of level 1 and 2 building 0 with varying $\varphi$. As the $\varphi$ increases, there are more interference RSS values ($r_i = 0.1 * (I - \min)$), which can increase the average localization error of RSS fingerprinting-based methods utilizing traditional machine learning model (KNN) and baseline deep learning model (CNN) as shown in Fig. 6.3.

## 6.4 The Architecture of ***SE-Loc***

This section introduces the system architecture of *SE-Loc*. The design of *SE-Loc* aims to secure fingerprinting-based indoor localization against potential AP attacks with signal-level and feature-level enhancements. To achieve the above goal, *SE-Loc* leverages four core modules to produce localization results with labeled and unlabeled RSS fingerprinting as the input data.

### 6.4.1 System Overview

Fig. 6.4 presents an overview of the proposed *SE-Loc*. To begin with, potential malicious APs may interfere RSS fingerprinting measurements. The **Normalization Module** (Section 6.3.3 and Section 6.4.3) normalizes RSS measurements and visualizes them by generating 2D fingerprint images. The **AP Selection Module** (Section 6.4.2) employs PCC for high-correlated AP selection, which improves the representation ability of RSS fingerprints. Then, the **Image Generation Module** (Section 6.4.3 and Section 6.5.3) further extrapolates fingerprints and generates fine-grained fingerprints image for more useful and robust feature extraction. At last, the **DAE-CNN Module** combines a denoising autoencoder with convolutional neural

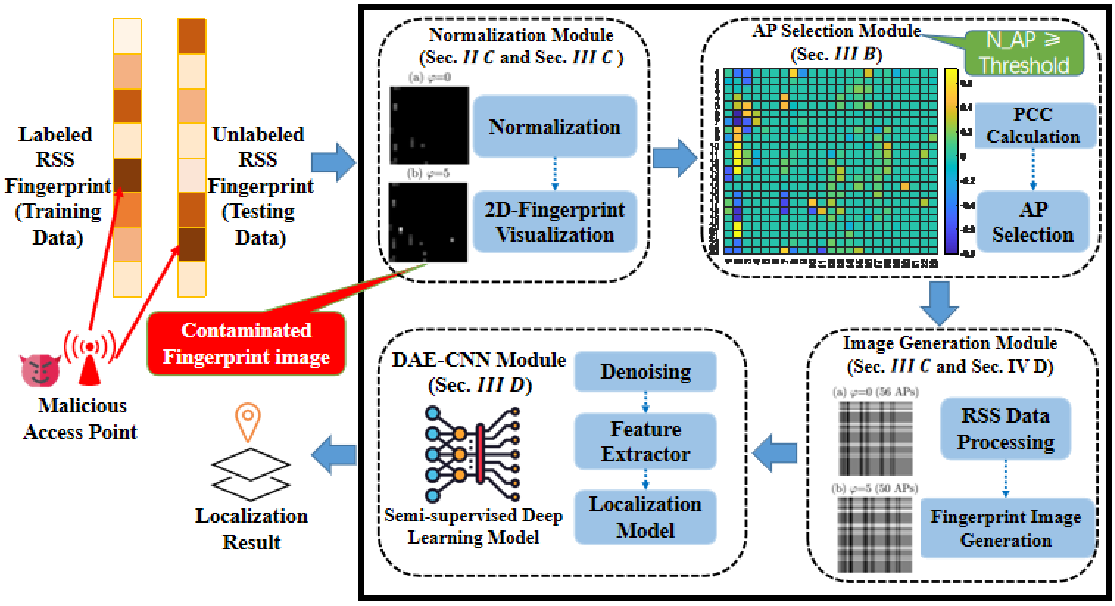

Labeled RSS Fingerprint (Training Data)
Unlabeled RSS Fingerprint (Testing Data)
Malicious Access Point
Contaminated Fingerprint image
Localization Result
Normalization Module (Sec. II C and Sec. III C )
(a) φ=0
(b) φ=5
Normalization
2D-Fingerprint Visualization
AP Selection Module (Sec. III B)
N_AP ≥ Threshold
PCC Calculation
AP Selection
Image Generation Module (Sec. III C and Sec. IV D)
(a) φ=0 (56 APs)
(b) φ=5 (50 APs)
RSS Data Processing
Fingerprint Image Generation
DAE-CNN Module (Sec. III D)
Denoising
Feature Extractor
Localization Model
Semi-supervised Deep Learning Model

**Figure 6.4 .** System Architecture of *SE-Loc*.

network.

In the offline phase, labeled RSS data is fed into the Normalization Module, AP Selection Module, Image Generation Module and DAE-CNN Module in turn. In particular, unlabeled RSS data is utilized to train the DAE model in DAE-CNN Module. After that, the DAE-CNN model is trained with both labeled and unlabeled RSS data. In the online phase, the above four modules are utilized together to process the unlabeled RSS data and find its best-match to predict corresponding location. Based on the above design, *SE-Loc* addresses the security and reliability issues of Wi-Fi fingerprint-based indoor localization simultaneously. The details of each module are introduced in corresponding Sections. This chapter presents the last three modules as follows.

### 6.4.2 Correlation-based AP Selection Module

In the real-world, an RSS fingerprint contains tens of and even hundreds of RSS measurements from surrounding APs. The computation complexity is significantly increased if all the RSS measurements from surrounding APs are processed. As a measure to deal with this issue, *Threshold* is denoted as the lower bound to trigger AP selection process. For instance, with the total number of surrounding AP in an indoor environment as $N_{AP}$, it becomes necessary to select APs for indoor localization when $N_{AP} >= Threshold$, and vice versa. It would be significantly difficult to identify malicious APs and exclude them from the fingerprinting data. Inspired by the previous observation that AP attacks always cause abrupt, random and fluctuant contaminations on top of RSS values of legitimate APs, this chapter introduces a correlation-based AP selection procedure in *SE-Loc*. The goal is to select the most stable, robust and mutual-correlated APs, while taming potential AP attacks and malicious interference on legitimate APs.

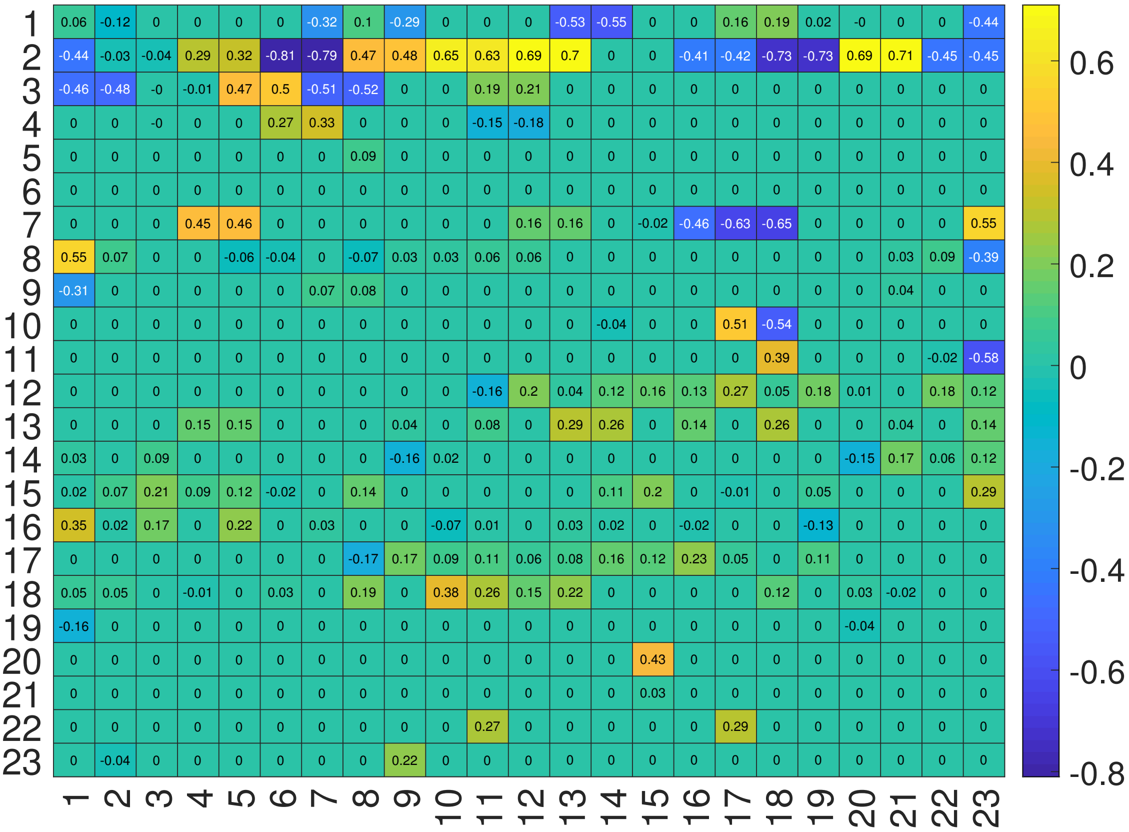
Assuming that in an industrial scenario, there are $N$ APs and $K$ reference points across the entire space. The normalized RSS value collected from the $i$th AP at the $k$th location is denoted by $r_i^k$. Therefore, this chapter uses a column vector $W_i = [r_i^1, r_i^2, \ldots, r_i^K]$ to represent the normalized RSS values of the $i$th AP across all $K$ locations. This chapter calculates the PCC [65] between each column vector $W_i$ and the location vector $\boldsymbol{l} = [l^1, l^2, \ldots, l^K]$, where $l^K$ is the location of the $K$th node. The equation can be derived by:

$$PCC_i = \frac{Cov(W_i, \boldsymbol{l})}{\delta_{W_i} \delta_{\boldsymbol{l}}}, \tag{6.4}$$

where $Cov(W_i, \boldsymbol{l})$ denotes the covariance between $W_i$ and $\boldsymbol{l}$, and $\delta$ denotes the standard deviation. The calculation result is a vector of correlation values denoted by $\mathbf{C}$. Note that the coefficient values range from -1 to 1, where a coefficient of -1 represents strong, negative correlation and a coefficient of 1 represents strong, positive correlation. Accordingly, a coefficient of 0 means that the input and output have no relationship [65]. Considering the magnitude of the correlation, if the RSS value of an AP has low correlation to the target location, it will not be useful to the fingerprints for localization. Therefore, this chapter chooses to select APs that have correlation values of target locations above a certain threshold (*e.g.*, $|PCC| >= 0.25$, derived experimental studies in Section 6.6). As shown in Fig. 6.5, APs whose correlation to the locations is over the threshold are selected. As a result, inconsequential RSS values of attacked APs and irrelevant APs can be removed from the RSS fingerprints. In addition, the correlation-based AP selection also helps to reduce computational load of *SE-Loc*'s semi-supervised deep learning model, improve localization accuracy, and reduce location prediction time.

### 6.4.3 Image Generation Module

To fully enhance the expressive ability of RSS fingerprints, this chapter designs an image generation module to convert RSS measurements into image-like fingerprints. Initially, the image generation module converts the above matrix into a gray image, where each element's value corresponds to the intensity of each pixel in the image. Note that the dimension of the 2D-fingerprint image is set as the square root of the closest squared value to the number of selected APs.



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.06 | -0.12 | 0 | 0 | 0 | 0 | -0.32 | 0.1 | -0.29 | 0 | 0 | 0 | -0.53 | -0.55 | 0 | 0 | 0.16 | 0.19 | 0.02 | -0 | 0 | 0 | -0.44 |
| 2 | -0.44 | -0.03 | -0.04 | 0.29 | 0.32 | -0.81 | -0.79 | 0.47 | 0.48 | 0.65 | 0.63 | 0.69 | 0.7 | 0 | 0 | -0.41 | -0.42 | -0.73 | -0.73 | 0.69 | 0.71 | -0.45 | -0.45 |
| 3 | -0.46 | -0.48 | -0 | -0.01 | 0.47 | 0.5 | -0.51 | -0.52 | 0 | 0 | 0.19 | 0.21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | -0 | 0 | 0 | 0.27 | 0.33 | 0 | 0 | 0 | -0.15 | -0.18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0.45 | 0.46 | 0 | 0 | 0 | 0 | 0 | 0 | 0.16 | 0.16 | 0 | -0.02 | -0.46 | -0.63 | -0.65 | 0 | 0 | 0 | 0 | 0.55 |
| 8 | 0.55 | 0.07 | 0 | 0 | -0.06 | -0.04 | 0 | -0.07 | 0.03 | 0.03 | 0.06 | 0.06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0.09 | -0.39 |
| 9 | -0.31 | 0 | 0 | 0 | 0 | 0 | 0.07 | 0.08 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.04 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.04 | 0 | 0 | 0.51 | -0.54 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.39 | 0 | 0 | 0 | -0.02 | -0.58 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.16 | 0.2 | 0.04 | 0.12 | 0.16 | 0.13 | 0.27 | 0.05 | 0.18 | 0.01 | 0 | 0.18 | 0.12 |
| 13 | 0 | 0 | 0 | 0.15 | 0.15 | 0 | 0 | 0 | 0.04 | 0 | 0.08 | 0 | 0.29 | 0.26 | 0 | 0.14 | 0 | 0.26 | 0 | 0 | 0.04 | 0 | 0.14 |
| 14 | 0.03 | 0 | 0.09 | 0 | 0 | 0 | 0 | 0 | -0.16 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.15 | 0.17 | 0.06 | 0.12 |
| 15 | 0.02 | 0.07 | 0.21 | 0.09 | 0.12 | -0.02 | 0 | 0.14 | 0 | 0 | 0 | 0 | 0 | 0.11 | 0.2 | 0 | -0.01 | 0 | 0.05 | 0 | 0 | 0 | 0.29 |
| 16 | 0.35 | 0.02 | 0.17 | 0 | 0.22 | 0 | 0.03 | 0 | 0 | -0.07 | 0.01 | 0 | 0.03 | 0.02 | 0 | -0.02 | 0 | 0 | -0.13 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.17 | 0.17 | 0.09 | 0.11 | 0.06 | 0.08 | 0.16 | 0.12 | 0.23 | 0.05 | 0 | 0.11 | 0 | 0 | 0 | 0 |
| 18 | 0.05 | 0.05 | 0 | -0.01 | 0 | 0.03 | 0 | 0.19 | 0 | 0.38 | 0.26 | 0.15 | 0.22 | 0 | 0 | 0 | 0 | 0.12 | 0 | 0.03 | -0.02 | 0 | 0 |
| 19 | -0.16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -0.04 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.27 | 0 | 0 | 0 | 0 | 0 | 0.29 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | -0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 0.22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 6.5 .** The Pearson Correlation Coefficient of 520 APs in building 0 level 2.

For instance, in Fig. 6.6, given 56 selected APs, the closest squared value is 64 and the dimension of the fingerprint image is 8 × 8. To fill up the image, 8 zero intensity are padded to the end of fingerprint vector [14]. In particular, 6 fingerprint images in Fig. 6.6 are captured at the same location, across different time under random AP attacks. It can be observed that the features of the above 6 images are distinctive from each other, showing that the normalized RSS vector cannot fully resist AP attacks.

To further enhance robustness and representativeness of the fingerprint image, this chapter enriches the characteristics of RSS data by introducing mutual difference, a new feature set in Section 6.5.3. The mutual differences of all selected

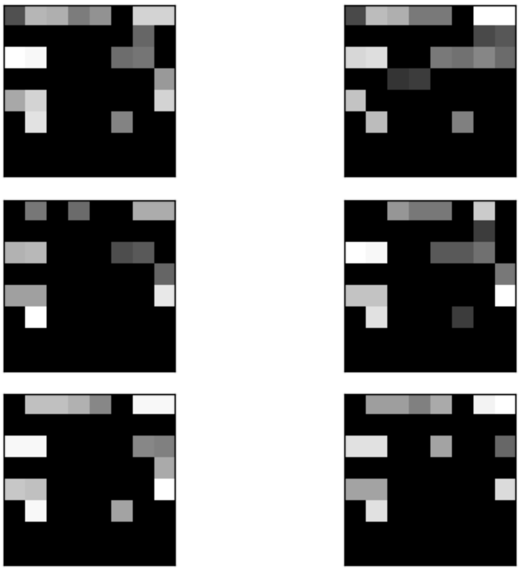


**Figure 6.6 .** The fingerprint images of normalized RSS vector from the selected 56 APs, captured at location $l_1$ at level 2, building 0, UJIIndoorLoc dataset.

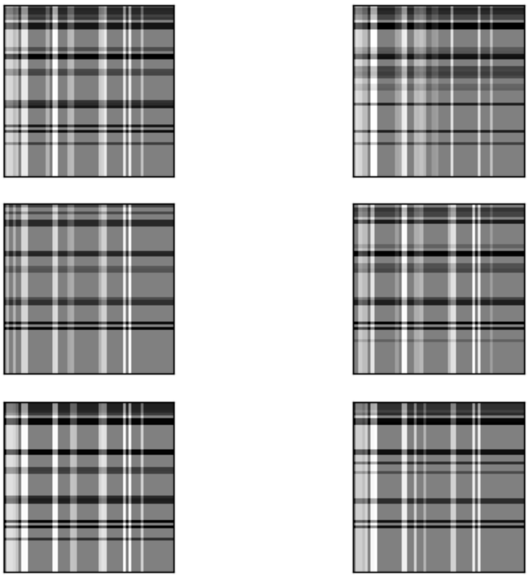
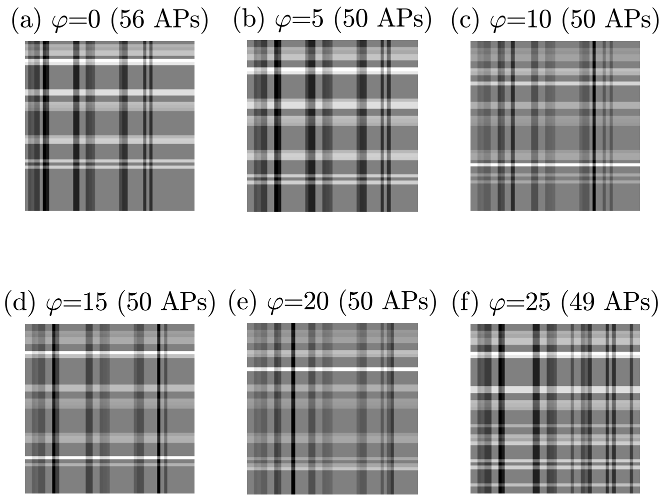


**Figure 6.7 .** The fingerprint images of processed RSS vector from the selected 56 APs, captured at location $l_1$ at level 2, building 0, UJIIndoorLoc dataset.

APs form a symmetric matrix, with all diagonal elements as 0. The fingerprint images based on mutual differences are shown in Fig. 6.7, where features are more fine-grained and stabilized.

This chapter further injects random AP attacks into RSS data by varying the number of attacking APs $\varphi$, where $\varphi = 5, 10, 15, 20, 25$. The observations from Fig. 6.8 validate that the newly generated fingerprint images are resilient to any scales of AP attacks, showing stable and robust features that will benefit the feature extraction and feature learning process of *SE-Loc*.



(a) $\varphi$=0 (56 APs) (b) $\varphi$=5 (50 APs) (c) $\varphi$=10 (50 APs)

(d) $\varphi$=15 (50 APs) (e) $\varphi$=20 (50 APs) (f) $\varphi$=25 (49 APs)

**Figure 6.8 .** The fingerprint images of processed RSS vector from the selected 56 APs, captured at location $l_1$ at level 2, building 0, UJIIndoorLoc dataset, by varying the number of attacks $\varphi$.

### 6.4.4 DAE-CNN Module for Feature Extraction with Semi-supervised Learning

This chapter designs a DAE-CNN module and extract useful features from RSS fingerprints for semi-supervised learning. The deep learning model of *SE-Loc* consists of a feature extractor based on DAE and a location predictor based on CNN.

***Feature Extraction Based on DAE***

Conventionally, the AE is a neural network model to learn efficient features from input data. However, if an autoencoder has more hidden layers than inputs, there is a risk for the autoencoder to learn the identity function (*e.g.*, the output simply equals the input), making the learning process become useless. To solve this issue, the Denoising Autoencoders (namely DAE) attempt to avoid this risk by introducing noise [111], so that the autoencoder must then 'denoise' or reconstruct the original input. For RSS fingerprinting data with malicious attacks, this chapter randomly corrupts the original RSS measurements with noises that are brought by random attacks.

The encoder of the AE model is a deterministic mapping function $f_\theta$ that transforms an input vector $\boldsymbol{X}$ into hidden representation $\boldsymbol{H}$. This chapter adopts the typical form for the encoder as an affine mapping with a non-linearity operation:

$$\boldsymbol{H} = f_\theta(\boldsymbol{X}) = f(\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}). \tag{6.5}$$

where $\boldsymbol{W}$ is a weight matrix, $\boldsymbol{b}$ is an offset vector and $\theta = \{\boldsymbol{W}, \boldsymbol{b}\}$ is the parameter set.

The hidden representation $\boldsymbol{H}$ is then mapped back to a reconstructed vector $\boldsymbol{Z}$. According to the input vector $\boldsymbol{X}$, there will be $\boldsymbol{Z} = g_{\theta'}(\boldsymbol{H})$. Here, the decoder of the AE model is a mapping function $g_{\theta'}$. Its typical form is again an affine mapping optionally followed by a squashing non-linearity:

$$\boldsymbol{Z} = g_{\theta'}(\boldsymbol{H}) = g(\boldsymbol{W'}\boldsymbol{H} + \boldsymbol{b'}), \tag{6.6}$$

where $\theta' = \{\boldsymbol{W'}, \boldsymbol{b'}\}$ is the parameter set.

Based on the above basic AE model, this chapter further proposed a DAE in *SE-Loc* by referring to [112]. The DAE model is designed to extract features and reconstruct the processed fingerprint images, which contain labeled and unlabeled RSS measurements from legitimate and malicious APs. In particular, as shown in

Fig. 6.9, after original processing, the RSS image $\boldsymbol{X}$ is corrupted to $\widetilde{\boldsymbol{X}}$ by

$$\widetilde{\boldsymbol{X}} = q(\boldsymbol{X}). \tag{6.7}$$

where the encoder of DAE is the mapping function from the corrupted RSS image $\widetilde{\boldsymbol{X}}$ to $\boldsymbol{H}$. Meanwhile, the decoder is another mapping function from $\boldsymbol{H}$ to $\boldsymbol{Z}$ by

$$\boldsymbol{H} = f_{\theta}(\widetilde{\boldsymbol{X}}) = f(\boldsymbol{W}\widetilde{\boldsymbol{X}} + \boldsymbol{b}), \boldsymbol{Z} = g_{\theta'}(\boldsymbol{H}) = g(\boldsymbol{W'}\boldsymbol{H} + \boldsymbol{b'}). \tag{6.8}$$

The final feature matrix extracted from the above DAE model is $\boldsymbol{H}$, which is then fed into the location prediction model.

### ***Location Prediction Model Based on CNN***

As shown in Fig. 6.9, the CNN model in *SE-Loc* performs location prediction tasks, which consists of the input layer, the convolution (Conv) layers, the pooling layers, the up-sampling layers, a fully connected layer and the output layer [5]. The input of CNN model is the feature matrix $H$ generated from the DAE model. The 2-dimensional convolutional operation in the CNN model of *SE-Loc* is described as follows.

$$x_j^{l^{(i)}} = \sum_{k=1}^{M} x_j^{l^{(i-1)}} \centerdot w_{kj}^{l^{(i)}} + b_j^{l^{(i)}}. \tag{6.9}$$

Here, notation $\centerdot$ is the convolution operator, matrix $x_j^{l^{(i-1)}}$ is the $j$th feature map of the previous layer $l^{(i-1)}$, and matrix $x_j^{l^{(i)}}$ is the $j$th feature map of the current layer $l^{(i)}$. In addition, $M$ is the number of feature map in the layer $l^{(i-1)}$, and $w_{kj}^{l^{(i)}}$ is the depth of kernel for the $j$th feature map of layer $l^{(i)}$. Note that $w_{kj}^{l^{(i)}}$ and $b_j^{l^{(i)}}$ are randomly initialized and initialized by zero, respectively. There are several types of nonlinear operations in the above CNN model and this chapter applies the ReLU for their optimizations.

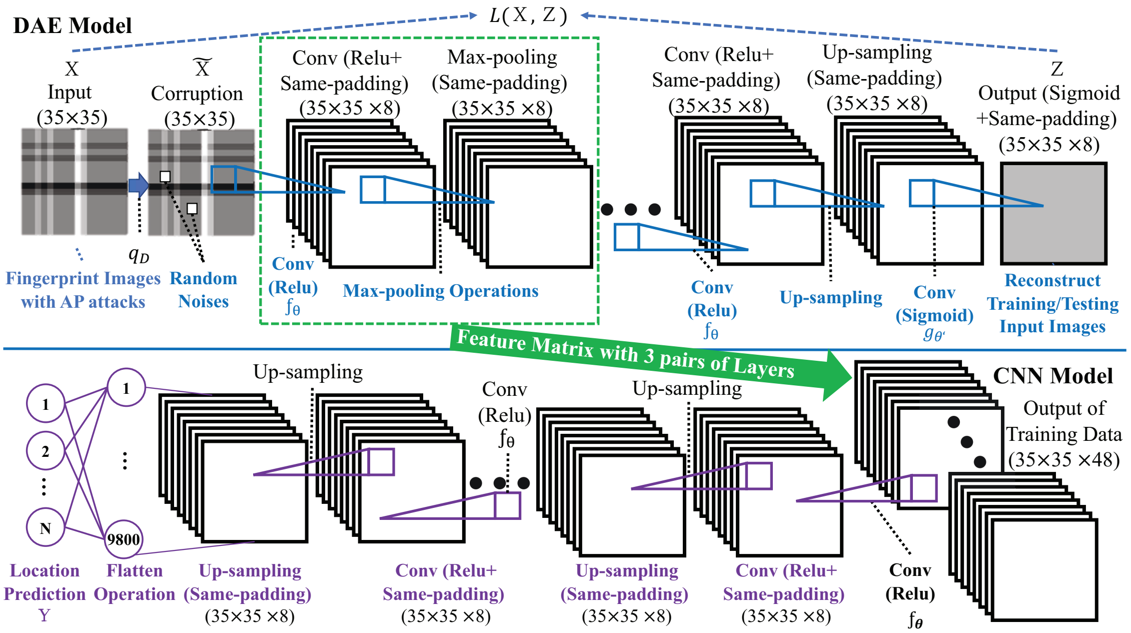

DAE Model
$L(X, Z)$
X
Input
(35×35)
$\widetilde{X}$
Corruption
(35×35)
$q_D$
Fingerprint Images with AP attacks
Random Noises
Conv (Relu+ Same-padding) (35×35 ×8)
Max-pooling (Same-padding) (35×35 ×8)
Conv (Relu) $f_\theta$
Max-pooling Operations
Conv (Relu+ Same-padding) (35×35 ×8)
Up-sampling (Same-padding) (35×35 ×8)
Z
Output (Sigmoid +Same-padding) (35×35 ×8)
Conv (Relu) $f_\theta$
Up-sampling
Conv (Sigmoid) $g_{\theta'}$
Reconstruct Training/Testing Input Images
Feature Matrix with 3 pairs of Layers
CNN Model
Output of Training Data (35×35 ×48)
1
2
N
1
9800
Up-sampling
Conv (Relu) $f_\theta$
Up-sampling
Location Prediction Y
Flatten Operation
Up-sampling (Same-padding) (35×35 ×8)
Conv (Relu+ Same-padding) (35×35 ×8)
Up-sampling (Same-padding) (35×35 ×8)
Conv (Relu+ Same-padding) (35×35 ×8)
Conv (Relu) $f_\theta$

**Figure 6.9 .** The semi-supervised learning model of DAE-CNN in *SE-Loc*.

In *SE-Loc*, multiple filters in CNNs are utilized to extract consecutive features at different levels. These features form the penultimate layer and are then passed to a fully connected Softmax layer. The final output is a probability distribution over a number of labels of reference points for localizations. The categorical cross-entropy is chosen as the loss function $l$ of the above location prediction process for optimization, which can be computed by

$$l = -\sum_{i=1}^{N} y_i log(p_i), \tag{6.10}$$

where $N$ is the number of classification labels. When $p_i$ equals the $i$th value of the Softmax output, $y_i = 1$; otherwise $y_i = 0$.

## 6.5 Implementation

For implementation, this chapter uses a Dell laptop with an Intel Core i7-7600U CPU as the system server of *SE-Loc*. The DAE-CNN model of *SE-Loc* is implemented on TensorFlow and Keras using Python 3.

### 6.5.1 BUPT Fingerprint Dataset Collection

In a practical Wi-Fi network, the malicious attacks and multi-path effects vary with the location of a mobile device. To verify the proposed approach to enhancing security and reliability of indoor localization, this chapter conducted a field experiment in a lab located at the main building of Beijing University of Posts and Telecommunications. As shown in Fig. 3.5, this chapter deploys TP-Link wireless routers as the Wi-Fi APs at different locations. For RSS measurement collection, this chapter employs a laptop equipped with a Phoenix Wi-Fi collector. At each reference point, the system server sequentially collects the corresponding RSS samples from all APs for 300 times. Overall, this chapter collected $33,600$ labeled RSS fingerprints, with $20,160$ as the training set and $13,440$ as the testing set.

### 6.5.2 UJIIndoorLoc Dataset and Extrapolated Fingerprints

To further verify the scalability of *SE-Loc*, this chapter further adopts a public Wi-Fi RSS fingerprint dataset called UJIIndoorLoc [87]. The indoor environment of UJIIndoorLoc covers 108,703 $m^2$ indoor areas across 3 adjacent buildings. Totally, 21,049 RSS measurements have been collected from 520 wireless APs, with 19,938 as the training set and the rest as the testing set. To enrich the scale of RSS fingerprints, this chapter reconstructs $M$ extrapolated copies and inject malicious APs in the new copies by introducing random fluctuations in RSS values with an upper limit ($\phi$) [91]. Correspondingly, in the extrapolated fingerprints, RSS vectors can have up to $\phi$ noise-induced values [14]. In the simulations, this chapter sets $M = 10$, $\phi = \{0, 10, 50, 100\}$. Correspondingly, there are 10 copies of each RSS fingerprints vector.

To determine the label of RSS fingerprints at each reference point, this chapter divides the localization area into a number of zones, *i.e.*, each zone is a grid area covering $12 \times 12$ $m^2$. To generate the label for each grid, this chapter adopts One-Hot Encoding [47] to map each grid into a One-Hot vector. Consequently, each individual grid represents a categorical variable, and the indoor localization task essentially becomes a classification problem across all grids, with the center of each grid as the reference point.

### 6.5.3 Data Processing

To enrich the characteristics of RSS data, this chapter adds a new feature set $\boldsymbol{R}$ to increase the dimension of fingerprints. $\boldsymbol{R}$ is the set of features that represent the mutual differences between the normalized RSS values of different APs. For instance, a basic entry of $r_i - r_j$ $(i, j \in APs)$ represents the difference between the normalized RSS value of AP $i$ and AP $j$. Together, the feature array of each individual AP together forms a feature matrix $\boldsymbol{R}$ as:

$$\boldsymbol{R} = \begin{bmatrix} 0 & r_1 - r_2 & r_1 - r_3 & \cdots & r_1 - r_n \\ r_2 - r_1 & 0 & r_2 - r_3 & \cdots & r_2 - r_n \\ r_3 - r_1 & r_3 - r_2 & 0 & \cdots & r_3 - r_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_n - r_1 & r_n - r_2 & r_n - r_3 & \cdots & 0 \end{bmatrix}. \tag{6.11}$$

According to Equation 6.1, the mutual differences between RSS values of AP $i$ and AP $j$ at the location $k$ can be calculated as:

$$\begin{aligned} RSS(d_i^k) - RSS(d_j^k) &= RSS(d_{0_i}^k) + 10\eta log(\frac{d_i^k}{d_{0_i}^k}) + n_{\sigma_i}^k + n_s \\ &- RSS(d_{0_j}^k) - 10\eta log(\frac{d_j^k}{d_{0_j}^k}) - n_{\sigma_j}^k - n_s \\ &= a + b * log(\frac{d_i^k}{d_j^k}) + \Delta n, \end{aligned} \tag{6.12}$$

where $b = 10\eta$, $a = RSS(d_{0_i}^k) - RSS(d_{0_j}^k) + blog(\frac{d_{0_j}^k}{d_{0_i}^k})$ and $\Delta n = n_{\sigma_i}^k - n_{\sigma_j}^k$. According to the above equation, the difference between the normalized RSS value from AP $i$ and AP $j$ can be derived by:

$$r_i - r_j = a' + b' * log(\frac{d_i^k}{d_j^k}) + \Delta n', \tag{6.13}$$

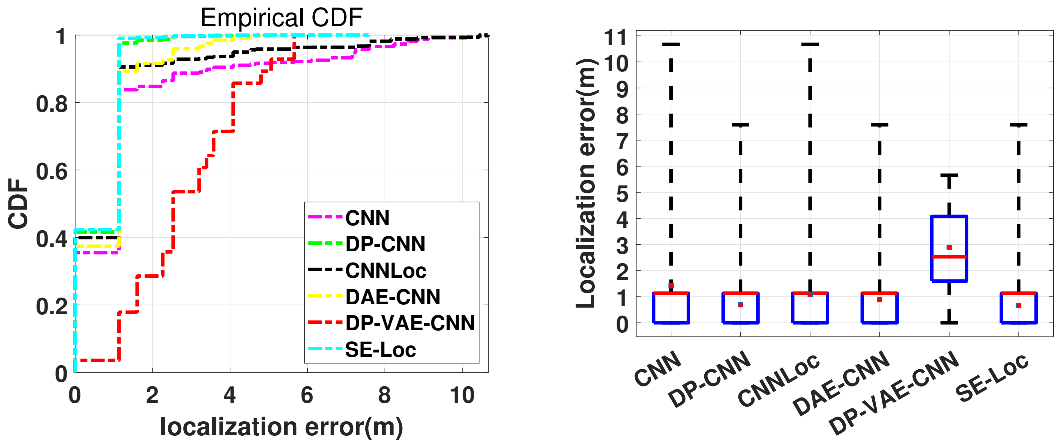
where $a' = \theta * a, b' = \theta * b$ and $\Delta n' = \theta * \Delta n$, $d_i^k$ is the range between the mobile user and AP $i$, $d_j^k$ is the range between the mobile user and the AP $j$. Consequently, the value of $\frac{d_i^k}{d_j^k}$ is dependent on the location of the mobile users once all APs are deployed. The above deduction suggests that the localization result is highly dependent on the feature matrix $R$, showing the feasibility of the data processing procedure in *SE-Loc*.

## 6.6 Experimental Study

This chapter conducts extensive experiments to evaluate: (a) the localization performance of *SE-Loc* under malicious AP attacks, (b) the effectiveness of AP selection component in *SE-Loc*, and (c) parameter tuning for the deep learning model, (d) ablation study.

### 6.6.1 Overall Performance Comparison with Baseline Methods

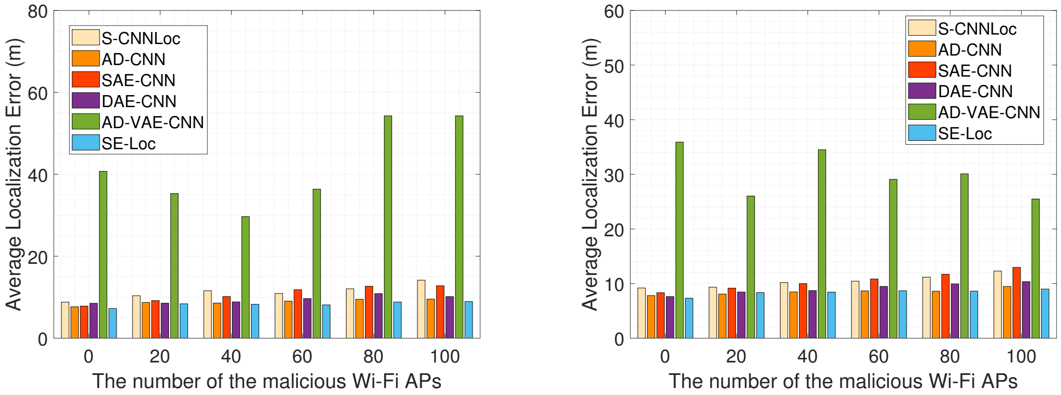
In evaluations, this chapter compares *SE-Loc* with the following state-of-the-art baseline methods. S-CNNLoc [14]: a CNN–based model with AP security-oriented fingerprint extrapolation. DP-CNN [65]: a CNN-based indoor localization method with fingerprint data processing. CNNLoc (SAE-CNN) [5]: a representative deep learning model that combines SAE and CNN for multi-building and multi-floor indoor localization. DAE-CNN: a CNN-based model combined with the DAE network. VAE-CNN [41]: a CNN-based model constructed with the VAE network. *SE-Loc*: a novel security-enhanced indoor localization model that this chapter constructs with DAE and CNN networks, combined with a rigorous AP selection component and fingerprint data processing. To fully evaluate the performance of *SE-Loc*, this chapter conducts indoor localization experiments with fingerprinting datasets from indoor scenarios of different scales. For the small-scale environment, this chapter leverages the BUPT dataset to evaluate the scalability of the above methods and the effectiveness of unsupervised learning by *SE-Loc*. For the large-scale environment, this chapter adopts the UJIIndoorLoc dataset and test the performance of indoor localization methods under different levels of AP attacks.



**(a)** The CDF distribution of localization errors **(b)** The box-plot distribution of localization errors

**Figure 6.10 .** Comparison of localization accuracy with different methods.

**Performance comparison on BUPT dataset:** The comparison of localization results on BUPT dataset is shown in Fig. 6.10a. Overall, *SE-Loc* outperforms all state-of-the-art baseline methods and achieves the highest indoor localization accuracy, where 99% of localization errors are lower than 2 $m$ and over 40% of localization results are within 1 $m$ to the groundtruth. The second-best performance is achieved by the DP-CNN method, showing that data processing plays an important role in extracting useful features from RSS fingerprints for deep learning models to learn. While the CNN-based method [65] only extracts high-level features to enhance localization accuracy, it guarantees 2 $m$ localization errors for only 80% of all testing fingerprinting samples. SAE-CNN [5] further improves the performance of CNN as it encodes raw RSS fingerprints by the SAE for its CNN network to learn more useful representations. The DP-VAE-CNN method performs the worst, where only 29% of all testing data is with localization error under 2 $m$.



**(a)** Level 1 **(b)** Level 2

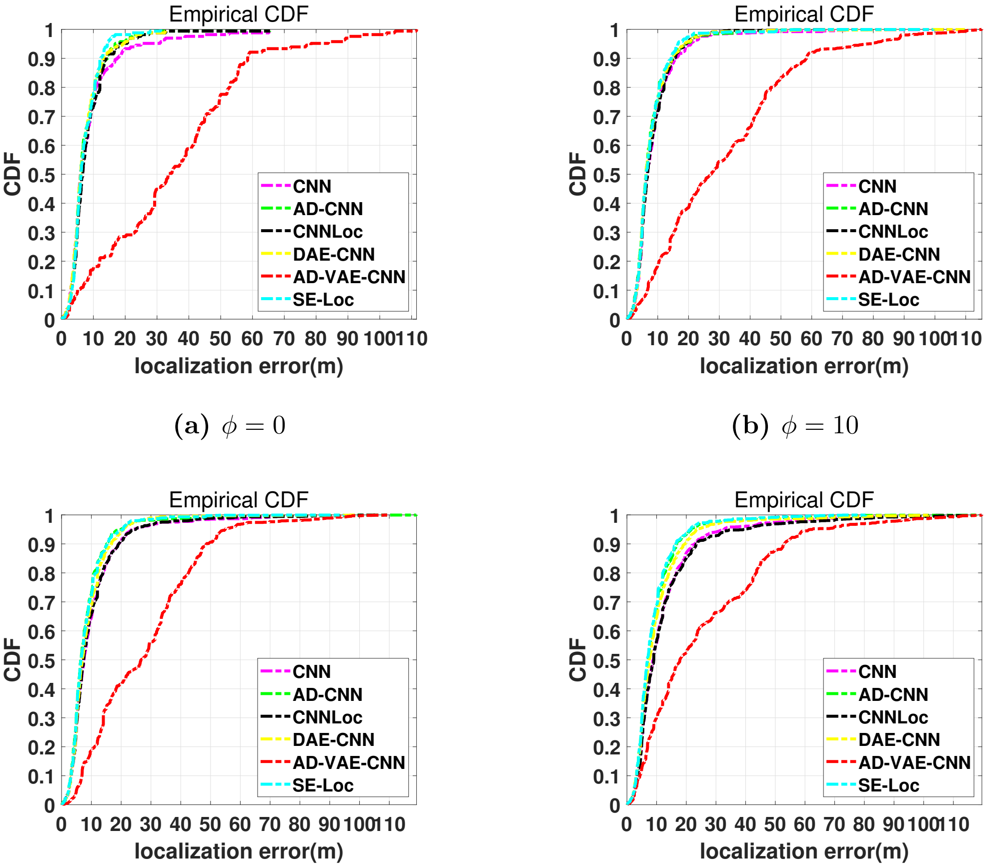
**Figure 6.11 .** The performance comparison of *SE-Loc* and the baseline methods in building 0, levels 1 and 2 of the UJIIndoorLoc dataset, with the upper limit of AP attacks varying from 0 to 100.

This chapter further draws a box-plot based on localization results on BUPT dataset in Fig. 6.10b. Compared with the CNN and SAE-CNN, *SE-Loc* has the smallest Inter-Quartile Range (IQR, *i.e.*, the distance between first quartile and third quartile) and the lowest median localization error of 0.66 $m$. Basically, CNN, DP-CNN, SAE-CNN and DAE-CNN models have similar performances, but with higher median errors than the method. The DP-VAE-CNN method performs the worst with the highest median errors of 2.9 $m$. The above results show that by selecting correlated APs and denoising RSS fingerprints with the DAE-CNN model, *SE-Loc* can successfully improve localization accuracy in the small-scale indoor environment.

**Performance comparison on UJIIndoorLoc dataset with AP attacks:** The localization results by different methods on UJIIndoorLoc dataset are presented in Fig. 6.11. In this experiment, this chapter sets $\phi$ as the upper limit of random

AP attacks and vary it from 0 to 20, 40, 60, 80 and 100. Here, an AP attack (*i.e.*, jamming or spoofing) changes the RSS value of a legitimate AP, as introduced in Equation 6.2. Accordingly, with the upper limit of attacked APs increased, the RSS fingerprint would become outliers to the groundtruth. In Fig. 6.11a, as the upper limit of AP attacks increases to 100, the average localization error of *SE-Loc* increases from 7.2 $m$ to 8.9 $m$ at level 1. However, the performance degradation of 1.7 $m$ is the smallest among all baseline methods. The corresponding error fluctuations of CNN, AD-CNN, CNNLoc and DAE-CNN methods are 5.4 $m$, 1.9 $m$, 5.0 $m$ and 2.4 $m$, respectively. Meanwhile, the highest localization error is up to 52.3 m when applying the AD-VAE-CNN model. Similar performance results are observed at level 2 in Fig. 6.11b, where *SE-Loc* remains resistant to any scales of malicious attacks on Wi-Fi APs in the localization system. Meanwhile, for CNN, CNNLoc, and DAE-CNN, the error fluctuations equal to 3.1 $m$, 4.7 $m$, and 3.3 $m$, respectively. Moreover, the error fluctuation of the AD-CNN method is stable, varying from 7.8 $m$ to 9.5 $m$ when $\phi$ increases from 0 to 100.

Fig. 6.12a to Fig. 6.12d further visualize the CDF of localization errors by different methods at levels 2, building 0, when the upper limit $\phi$ of random AP attacks increases from 0 to 10, 50 and 100. The four CDF results show that *SE-Loc* maintains over 80% and 70% of localization errors within the errors of 10 $m$, despite that there are $\phi = 50$ and $\phi = 100$ random AP attacks. The 10% performance degradation also outperforms all baseline methods, which validates the effectiveness of the security enhancement of *SE-Loc* for indoor localization. The above comparison results demonstrate that *SE-Loc* has superior resilience to different scales of AP



**(a)** $\phi = 0$

**(b)** $\phi = 10$

**(c)** $\phi = 50$

**(d)** $\phi = 100$

**Figure 6.12 .** The Localization performance of the proposed method as compared to other methods on UJIIndoorLoc dataset of building 0 level 2 with a varying upper limit of malicious APs.

attacks and is reliable to deliver stable and effective indoor localization results.

### 6.6.2 Evaluation of the AP Selection Component

Next, this chapter evaluates *SE-Loc*'s AP selection component. Note that the UJIIndoorLoc dataset contains RSS measurements collected from 520 APs, but only part of them are highly correlated, as this chapter has introduced in Section 6.4.2.

For indoor localization systems under malicious AP attacks, the AP selection process tackles contaminated APs and data redundancy simultaneously.

To test the effectiveness of the AP selection component, this chapter conducted a series of experiments, and the results are summarized in Table 6.1. Here, the key factor is the threshold of Pearson Correlation Coefficient, which indicates the correlation value of APs that can be chosen to form RSS fingerprints. This chapter sets the threshold's value from 0 (*e.g.*, DAE-CNN is without AP selection component) to 0.1, 0.15, 0.2, 0.25 and 0.3.

As shown in Table 6.1, when the PCC threshold decreases from 0.3 to 0.1, the numbers of selected APs at levels 1, 2 and 3 increase from 41 to 89, 45 to 103 and 34 to 83, respectively. Correspondingly, the average location prediction time increases from 0.55 $ms$ to 0.63 $ms$, as more APs are involved in the computation process. Compared with the DAE-CNN method, which has no AP selection process, *SE-Loc* achieves more stable and accurate localization results with lower errors for most levels of building 0. The above results validate that it is essential for indoor localization systems to combine the AP selection component, since AP selection helps to reduce localization time and improves localization accuracy even under potential AP attacks. Considering a trade-off between localization accuracy and location prediction time, this chapter highlights a gray column that features the most suitable PCC threshold (*i.e.*, 0.25) for *SE-Loc*. In the following experiments, this chapter sets the PCC value of *SE-Loc* model as 0.25.

Table 6.1 : Evaluation on the AP selection component with different PCC values at building 0, UJIIndoorLoc dataset.

| Methods | PCC Threshold | Localization Error based on UJIIndoorLoc dataset of building 0 (m) | | | Number of Selected APs | | | Average prediction time (ms) |
|---|---|---|---|---|---|---|---|---|
| | | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 | |
| DAE-CNN + AP Selection | 0.3 | 8.23 | 8.05 | 7.99 | 41 | 45 | 34 | 0.55 |
| | 0.25 | 7.84 | 7.85 | 7.94 | 50 | 56 | 39 | 0.57 |
| | 0.2 | 7.48 | **7.62** | 7.84 | 61 | 63 | 46 | 0.58 |
| | 0.15 | 7.41 | 7.9 | 8.57 | 69 | 85 | 57 | 0.6 |
| | 0.1 | **7.29** | 7.65 | **7.36** | 89 | 103 | 83 | 0.63 |
| DAE-CNN | 0 | 8.51 | 7.9 | 8.9 | 520 | | | 1.43 |

### 6.6.3 Parameter Tuning and Ablation Study for *SE-Loc*'s Model

This chapter further evaluates *SE-Loc*'s model (*i.e.*, DP-DAE-CNN) by tuning different parameters. The key parameters for *SE-Loc* include the number of Conv layers in DAE, the number of Conv layers in CNN, and the number of filters (*e.g.*, kernels) in all Conv layers. For example, a parameter set 3-5-8 equals to a *SE-Loc* model with 3 DAE layers, 5 CNN layers (including a fully connected Softmax layer) and 8 filters in all Conv layers. Meanwhile, all the other parameters are set as follows: the kernel size is $2 \times 2$, the stride value is 1, the max-pooling is by $2 \times 2$ and the up-sampling is by $1 \times 1$.

***Impact of Model Parameters on the Localization Performance in BUPT Dataset***

First, this chapter focuses on the number of filters in all convolution layers. According to the first three groups of bars in Fig. 6.13, when the number of filters increases from 8 to 16 and 32, the localization accuracy improves for all compared methods. The average localization errors decrease from 1.1 $m$ to 0.7 $m$, 0.76 $m$ to 0.66 $m$ and 0.7 $m$ to 0.65 $m$ by the DAE-CNN, DP-CNN and DP-DAE-CNN (*i.e.*, *SE-Loc*), respectively. The above results reflect the negative correlation between the number of filters and the average localization error. Second, this chapter evaluates the impact of DAE layers on the localization results. According to the 2nd, 4th and 5th groups of bars, when the number of Conv layers in DAE increases from 3 to 4 and 5 while the other parameters keep the same, the localization errors show a slight rise for DAE-CNN while staying the same level for DP-CNN and DP-DAE-CNN.

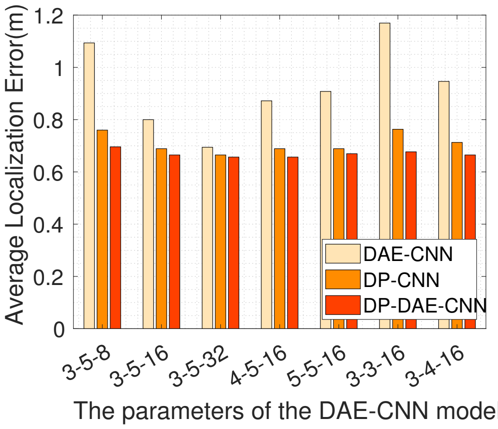


**Figure 6.13 .** The average localization errors by different methods with different model parameters in BUPT dataset.

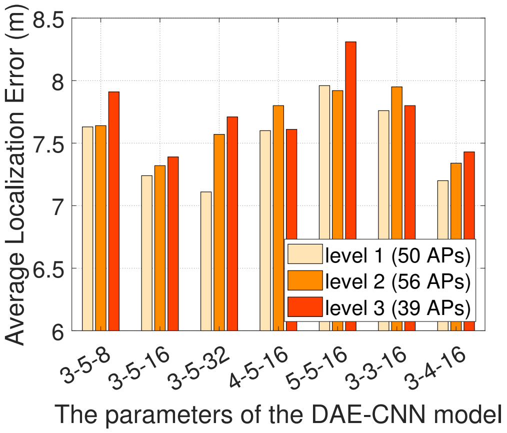


**Figure 6.14 .** The average localization errors by tuning parameters of *SE-Loc* at all levels of building 0, UJIIndoorLoc dataset.

Third, this chapter studies the impact of number of layers in CNN. Based on the 6th, 7th and 2nd groups of bars, the average localization errors decrease significantly with fewer Conv layers in CNN.

### *Impact of Model Parameters on the Localization Performance in UJI-IndoorLoc Dataset*

Fig. 6.14 shows the average localization errors by the proposed DAE-CNN model with different parameters at levels 1, 2 and 3, building 0 in the UJIIndoorLoc dataset. The key parameters are the same, including the number of Conv layers in DAE, the number of Conv layers in CNN, and the number of filters (*e.g.*, kernels) in all Conv layers. In the first three groups of bars, as the number of filters in all Conv layers increases from 8 to 16 and 32, the average localization errors by *SE-Loc* fluctuate

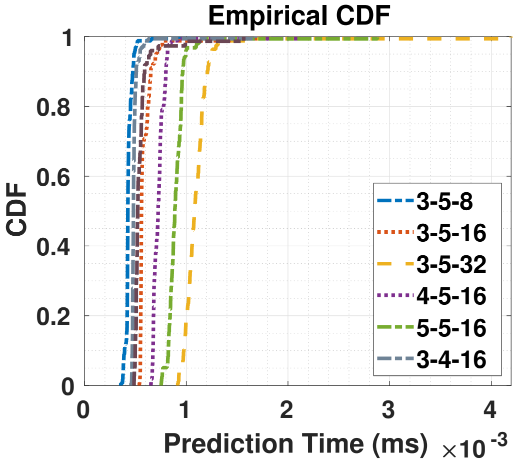
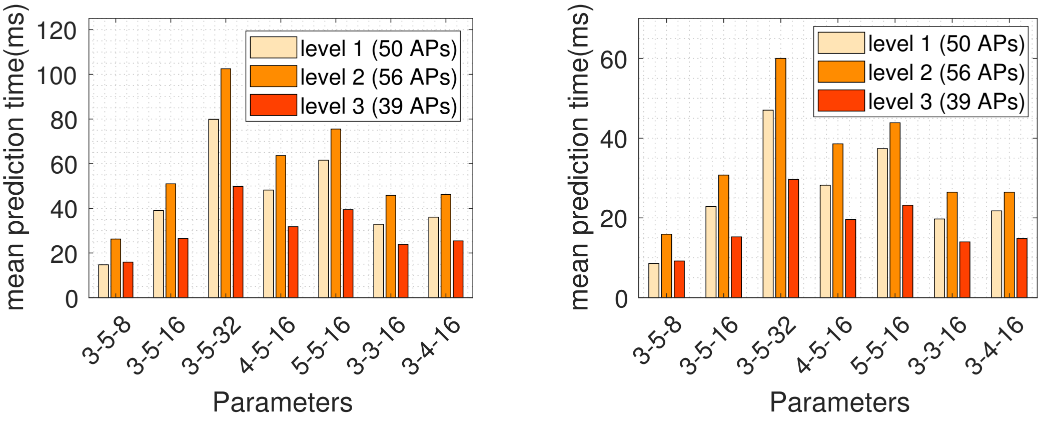


**Figure 6.15 .** The CDF of prediction time with different model parameters in BUPT dataset via the Raspberry Pi.

from 7.6 $m$ to 7.1 $m$, 7.6 $m$ to 7.3 $m$ and 7.9 $m$ to 7.4 $m$ at three levels, respectively. It can also be observed from the 2nd, 4th, and 5th groups of bars that the localization performance of *SE-Loc* shows a decline when the DAE has more layers. According to the 2nd, 6th, and 7th groups of bars, the average localization improves slightly at all levels as the number of layers in CNN model increases from 3 to 5. The above results show that by having more layers in the CNN model, *SE-Loc* can extract better representations from RSS fingerprints with AP attacks, thus enhancing the reliability and accuracy of indoor localization.

### *Impact of Model Parameters and Device Heterogeneity on the Mean Prediction Time*

For ILBSs providers, location prediction time (*i.e.*, computation time of localization) is of great importance in system performance. In this subsection, this chapter conducts extensive experiments to evaluate the impact of model parameters and de-



**(a)** The mean prediction time on Raspberry Pi **(b)** The mean prediction time on the Redmi K30

**Figure 6.16 .** The mean prediction time by tuning parameters of *SE-Loc* at all levels of building 0, UJIIndoorLoc dataset via different devices.

vice heterogeneity on the mean prediction time of *SE-Loc*. First, this chapter varies the model parameter of *SE-Loc* and conduct localization experiments with a Raspberry Pi using BUPT dataset. Overall, the CDF curves in Fig. 6.15 show that the Raspberry Pi runs *SE-Loc* within $1ms$ for most parameter settings. The prediction time slightly increases when more layers are added to each component of *SE-Loc*'s model. Second, this chapter further evaluates the impact of device heterogeneity on the mean prediction time of *SE-Loc* in Fig. 6.16. This chapter conducts experiments by varying the model parameters and using different devices on the UJIIndoorLoc dataset. Fig. 6.16a, Fig. 6.16b and Fig. 6.17 present the experimental results with a Raspberry Pi, a Redmi K30 and a Dell laptop, respectively. It can be observed that for *SE-Loc* model with the same parameter, the mean prediction times on different

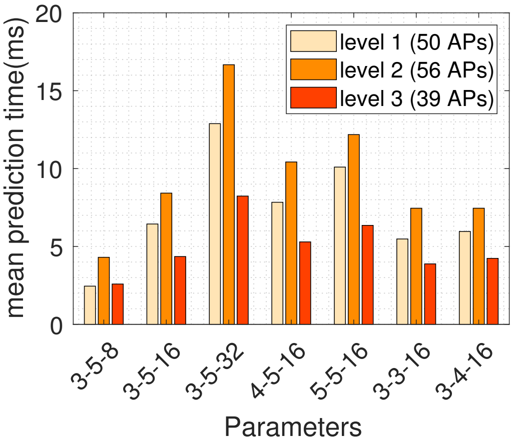


**Figure 6.17 .** The mean prediction time by tuning parameters of *SE-Loc* at all levels of building 0, UJIIndoorLoc dataset on the server.

devices are similar and stable. Moreover, *SE-Loc* shows more efficiency in prediction time when it has less complex model parameters.

The above results validate that *SE-Loc* is robust to device heterogeneity, meanwhile, its model parameters can make trade-off with the mean prediction time. Combing the experimental results in Fig. 6.14, the *SE-Loc* model with 3 Conv layers in DAE, 5 Conv layers in CNN, and 16 filters in all Conv layers can achieve the best trade-off between localization accuracy and mean prediction time on heterogeneous devices, with the localization error of 0.65 $m$ and mean prediction time of 25 $ms$.

### *Ablation Study on SE-Loc*

To illustrate the effectiveness of each model component, this chapter further conducts an ablation study with different variants of *SE-Loc* as follows. Table 6.2 presents the localization performance of *SE-Loc* together with its three variants on

Table 6.2 : Comparison of localization errors by different methods based on two types of datasets

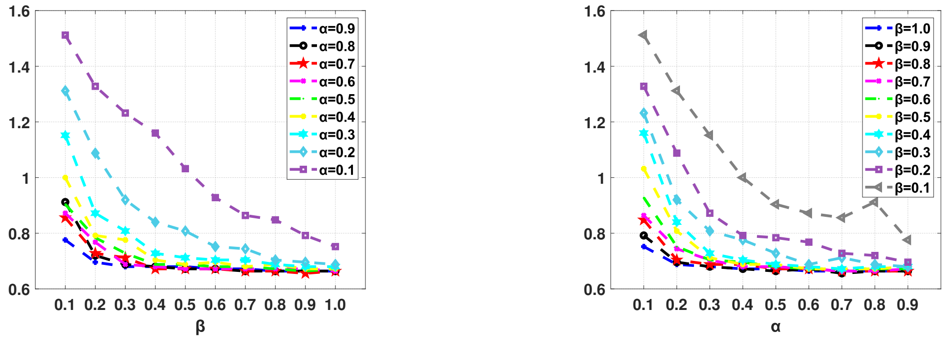
| Methods | Localization error (m) on UJIIndoorLoc dataset | | | Methods | Localization error (m) on BUPT dataset |
|---|---|---|---|---|---|
| | level 1 | level 2 | level 3 | | |
| DAE-CNN | 8.51 | 7.9 | 8.9 | DAE-CNN | 0.8 |
| AP Selection + DAE-CNN | 7.84 | 7.85 | 7.94 | N/A | – |
| AP Selection + Data Processing + CNN | 7.67 | 7.79 | 7.87 | Data Processing + CNN | 0.69 |
| AP Selection + Data Processing + DAE-CNN | **7.24** | **7.32** | **7.62** | Data Processing + DAE-CNN | **0.66** |

the UJIIndoorLoc dataset and BUPT dataset, respectively. On the left side of Table 6.2, the variant DAE-CNN shows the largest errors across all levels. Meanwhile, DAE-CNN with the AP selection component or/and the data processing component can achieve more accurate localization results. In addition, when the DAE is removed from *SE-Loc*, the localization error increases significantly. On the right side of Table 6.2, this chapter compares compare two variants with *SE-Loc*. Note that for BUPT dataset, this chapter skips the AP selection component as the total number of APs is limited. Similar to localization results on UJIIndoorLoc, *SE-Loc* outperforms other variants in the BUPT dataset by achieving the smallest average localization error (*i.e.*, 0.66 $m$). Without the data processing component or the DAE component, the performance of *SE-Loc*'s variants degrades to different degrees. The above ablation study validates the effectiveness of each component of *SE-Loc*, showing that each component has its own contribution to improving localization accuracy.

### 6.6.4 Evaluation of Semi-Supervised Learning of *SE-Loc*

Finally, this chapter evaluates the effectiveness of semi-supervised learning of *SE-Loc* with BUPT dataset. As introduced in Sec. 6.4.4, *SE-Loc* is integrated with a denoising autoencoder that can extract features from RSS fingerprints with attacking APs and reconstruct original fingerprinting data. Let $\alpha$ and $\beta$ denote the size of training set and the ratio of labeled samples in the training set.

Fig. 6.18a shows the average localization error by *SE-Loc* by varying the ratio of labeled samples. When $\beta$ increases from 10% to 100%, the localization error



**(a)** Average localization error with respect to $\beta$ **(b)** Average localization error with respect to $\alpha$

**Figure 6.18 .** Average localization error with different sizes of the labeled set in the training data on BUPT dataset.

continues to decrease for all values of $\alpha$. For example, when $\alpha = 0.1$, the localization error is reduced from 1.5 $m$ to 0.75 $m$ by 50%. When $\beta = 0.6$, the localization results are all within 1 $m$ to the groundtruth regardless of the size of the training set, showing that *SE-Loc* can achieve satisfactory semi-supervised indoor localization with only 60% of training data labeled.

Fig. 6.18b presents the localization error of *SE-Loc* with respect to $\alpha$. As the value of $\alpha$ increases from 0.1 to 0.9, *SE-Loc* shows performance improvements with smaller average localization errors. When $\alpha = 0.5$, with only 50% of data as the training set, *SE-Loc* can still achieve a small localization error (*i.e.*, less than 0.8 $m$) in most cases. Fig. 6.19 presents the localization results by varying the size of training set for *SE-Loc* with different model parameters. When the parameters are set as 3-5-8 (*i.e.*, 3 Conv layers in DAE, 5 Conv layers in CNN, and 8 filters in all Conv layers), the average localization error decreases sharply as $\alpha$ increases from

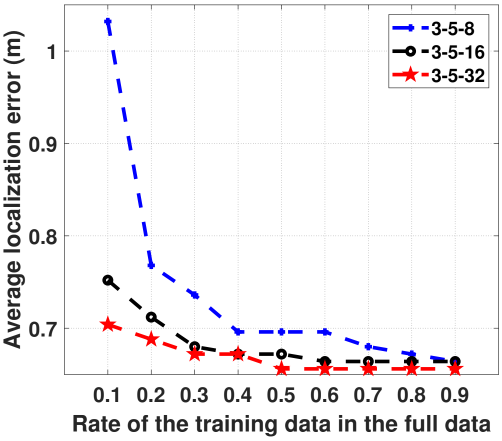


**Figure 6.19 .** Average localization error *vs.* different model parameters and different sizes of training set.

0.1 to 0.4. When the parameters are set as 3-5-16 and 3-5-32, similar performance is observed. For $\alpha$ with values larger than 0.6, *SE-Loc* can only slightly improve the localization results.

## 6.7 Conclusion

This chapter proposes *SE-Loc*, a deep learning-based technique to overcome the security vulnerabilities of indoor localization under random and unpredicted AP attacks. To achieve both reliable and secure localization, *SE-Loc* features both signal-level and feature-level enhancements simultaneously. In particular, this chapter combines a denoising autoencoder with convolutional neural networks for effective feature extraction and effective representation learning on RSS fingerprints. Extensive experiments have demonstrated that when confronting AP attacks, *SE-Loc* has superior performance over the baseline methods.

# Chapter 7

# Thesis Conclusions and Future Works

## 7.1 Summary of Thesis

In this thesis, Wi-Fi RSS fingerprinting-based indoor localization utilizing deep learning is analyzed and studied. In Chapter 1, an overview of indoor localization of IoT applications (covering the LBS of IoT applications, signals and measurements) and deep learning-based methods (mainly supervised and semisupervised) is presented. Secondly, the main motivations and challenges in indoor localization are discussed, including multipath effects and noise, environment dynamic, database problem, device heterogeneity, energy efficiency, and privacy and security. In Chapter 2, the existing works of the traditional indoor localization and promising deep learning methods are reviewed. After that, four Wi-Fi RSS fingerprinting-based indoor localization utilizing deep learning are presented from Chapter 3 to Chapter 6, respectively. In this chapter, the technical contributions and future work of this thesis are presented.

## 7.2 Summary of Contributions

The main contributions of this thesis are listed as follows:

(1) Firstly, RSS data are not only prone to multi-path reflections but also sensitive to time-varying environmental dynamics, which are the basic challenges

of indoor localization. In contrast to existing solutions focusing on spatial features of RSS, a novel indoor localization method is proposed by exploiting the temporal dependency of RSS time-series data and integrating the Kalman filter with DNNs. Extensive field experiments are conducted with a real-world testbed, and the experiment results validate the effectiveness of the Kalman-DNN method.

(2) Secondly, CapsLoc, a robust indoor localization system utilizing CapsNet is proposed, which can achieve high localization accuracy with Wi-Fi RSS fingerprinting. Specifically, hierarchical structures in Wi-Fi RSS fingerprinting can be efficiently extracted by the CapsNet model to address the typical challenges, such as the multipath effects and noise, the environment dynamics, and data limitation. The experimental results show that CapsLoc can realize higher localization performance over traditional machine learning methods (KNN and SVM) and existing deep learning methods (CNN and SAE-CNN).

(3) Thirdly, RSS data collected at the heterogeneous devices are time-varying when facing the challenges such as the device heterogeneity and database problem together with other classic challenges with wireless communications. To address these significant challenges, EdgeLoc, a robust and real-time indoor localization system towards heterogeneous IoT devices is presented. Extensive field experiment results show that EdgeLoc outperforms the state-of-the-art SAE-CNN method in localization accuracy.

(4) Finally, *SE-Loc*, a deep learning-based technique to enhance resiliency and

security of wireless indoor localization is presented to address the security challenge in Wi-Fi indoor localization. The architecture of *SE-Loc* consists of two parts: (1) AP selection for processing initially contaminated APs, and (2) a deep learning model based on a DAE and CNN for feature learning and location estimation. Extensive experiments show that *SE-Loc* outperforms the baseline methods on secured indoor localization. When there are up 100 malicious attacking APs in the UJIIndoorLoc database, *SE-Loc* can still realize a low error fluctuation and lowest average localization error compared to the state-of-the-art baselines.

## 7.3 Future Works

The future work includes designing and developing ultra-light and low-cost localization algorithms for IoT devices, such as IoT-enabled lighting bubbles and coin-battery-powered IoT devices [113]. These IoT devices have limited computation power and constrained battery capacity, so that energy and computation aware localization solutions are the keys to success. In recent years, efficient neural networks that may have the ability of achieving high performance on various types of edge devices include SqueezeNet[114], MobileNets[115], ShuffleNet[116], MobileNetv2[117], PeleeNet[118], ShuffleNetv2[119], MnasNet[120], MobileNetv3[121] and MobileViT[122] etc.. A possible solution may be allocating heavy tasks to the edge server and letting IoT devices run ultra-lighting jobs [101].

In addition to the heterogeneity of IoT devices, the vulnerability of Wi-Fi APs may lead to various security threats to ILBS users [14]. Although malicious AP attacks are taken into consideration in this thesis, there are various other attacks. With the vulnerability of Wi-Fi signals at both mobile devices and APs [123], other various attacks can be conducted towards wireless indoor localization, for example, AP hijacking, jamming, and man-in-the-middle attack [124]. These attacks not only degrade the performance of indoor localization but also introduce significant risks to the confidential location data of valuable and critical IoT assets [92]. Therefore, it is of great significance to take the vulnerabilities of indoor localization systems into consideration and develop security-enhanced deep learning techniques to protect indoor localization from the above attacks.

# Bibliography

[1] Q. Ye, X. Fan, G. Fang, H. Bie, Exploiting temporal dependency of RSS data with deep learning for IoT-oriented wireless indoor localization, Internet Technology Letters e366doi:https://doi.org/10.1002/itl2.366.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.366

[2] Q. Ye, X. Fan, G. Fang, H. Bie, X. Song, R. Shankaran, Capsloc: A robust indoor localization system with Wi-Fi fingerprinting using capsule networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–6.

[3] Q. Ye, H. Bie, K.-C. Li, X. Fan, L. Gong, X. He, G. Fang, Edgeloc: A robust and real-time localization system toward heterogeneous IoT devices, IEEE Internet of Things Journal 9 (5) (2022) 3865–3876. doi:10.1109/JIOT.2021.3101368.

[4] Q. Ye, X. Fan, H. Bie, D. Puthal, T. Wu, X. Song, G. Fang, Se-loc: Security-enhanced indoor localization with semi-supervised deep learning, IEEE Transactions on Network Science and Engineering (2022) 1–1doi:10.1109/TNSE.2022.3174674.

[5] X. Song, X. Fan, X. He, C. Xiang, Q. Ye, X. Huang, G. Fang, L. L. Chen, J. Qin, Z. Wang, Cnnloc: Deep-learning based indoor localization with Wi-Fi

fingerprinting, in: 2019 IEEE Ubiquitous Intelligence and Computing(UIC), 2019.

[6] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, G. Fang, A novel convolutional neural network based indoor localization framework with Wi-Fi fingerprinting, IEEE Access 7 (2019) 110698–110709.

[7] S. Lai, X. Fan, Q. Ye, Z. Tan, Y. Zhang, X. He, P. Nanda, Fairedge: A fairness-oriented task offloading scheme for IoT applications in mobile cloudlet networks, IEEE Access 8 (2020) 13516–13526.

[8] Y. Li, Y. Zhuang, X. Hu, Z. Gao, J. Hu, L. Chen, Z. He, L. Pei, K. Chen, M. Wang, X. Niu, R. Chen, J. Thompson, F. M. Ghannouchi, N. El-Sheimy, Toward location-enabled IoT (LE-IoT): IoT positioning techniques, error sources, and error mitigation, IEEE Internet of Things Journal 8 (6) (2021) 4035–4062. `doi:10.1109/JIOT.2020.3019199`.

[9] N. Singh, S. Choe, R. Punmiya, Machine learning based indoor localization using Wi-Fi RSSI fingerprints: An overview, IEEE Access 9 (2021) 127150–127174. `doi:10.1109/ACCESS.2021.3111083`.
URL `https://ieeexplore.ieee.org/ielx7/6287639/9312710/09531633.pdf?tp=&arnumber=9531633&isnumber=9312710&ref=`

[10] A. Billa, I. Shayea, A. Alhammadi, Q. Abdullah, M. Roslee, An overview of indoor localization technologies: Toward IoT navigation services, in: 2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT), 2020, pp. 76–81. `doi:10.1109/ISTT50966.2020.9279369`.

[11] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, J. Shang, Indoor localization improved by spatial context—A survey, ACM Comput. Surv. 52 (3) (Jul. 2019). doi:10.1145/3322241.
URL https://doi.org/10.1145/3322241

[12] M.-D. Kim, J. Liang, J. Lee, J. Park, B. Park, Path loss measurements and modeling for indoor office scenario at 28 and 38 GHz, in: 2016 International Symposium on Antennas and Propagation (ISAP), 2016, pp. 64–65.

[13] P. Tiwary, A. Pandey, S. Kumar, Differential d-vectors for RSS based localization in dynamic IoT networks, in: 2021 International Conference on COMmunication Systems NETworkS (COMSNETS), 2021, pp. 82–85. doi:10.1109/COMSNETS51098.2021.9352896.

[14] S. Tiku, S. Pasricha, Overcoming security vulnerabilities in deep learning-based indoor localization frameworks on mobile devices, ACM Transactions on Embedded Computing Systems 18 (6) (2020) 1–24.

[15] B. Wang, Q. Chen, L. T. Yang, H.-C. Chao, Indoor smartphone localization via fingerprint crowdsourcing: challenges and approaches, IEEE Wireless Communications 23 (3) (2016) 82–89. doi:10.1109/MWC.2016.7498078.

[16] S. Khodayari, M. Maleki, E. Hamedi, A RSS-based fingerprinting method for positioning based on historical data, in: Proceedings of the 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '10), 2010, pp. 306–310.

[17] S. Bozkurt, G. Elibol, S. Gunal, U. Yayan, A comparative study on machine learning algorithms for indoor positioning, in: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), 2015, pp. 1–8. doi:10.1109/INISTA.2015.7276725.

[18] S. B. A. Khattak, M. A. Pasha, M. U. Farooq, N. U. Hassan, C. Yuen, Empirical performance evaluation of Wi-Fi fingerprinting algorithms for indoor localization, in: 2018 IEEE International Conference on Communication Systems (ICCS), 2018, pp. 303–308. doi:10.1109/ICCS.2018.8689182.

[19] Z. Wu, K. Fu, E. Jedari, S. R. Shuvra, R. Rashidzadeh, M. Saif, A fast and resource efficient method for indoor positioning using received signal strength, IEEE Transactions on Vehicular Technology 65 (12) (2016) 9747–9758.

[20] R. H. Jaafar, S. S. Saab, A neural network approach for indoor fingerprinting-based localization, in: 2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), 2018, pp. 537–542. doi:10.1109/UEMCON.2018.8796646.

[21] P. Roy, C. Chowdhury, A survey of machine learning techniques for indoor localization and navigation systems, Journal of Intelligent & Robotic Systems 101 (3) (2021) 63. doi:10.1007/s10846-021-01327-z.
URL https://doi.org/10.1007/s10846-021-01327-zhttps://link.springer.com/content/pdf/10.1007/s10846-021-01327-z.pdf

[22] C. Liu, C. Wang, J. Luo, Large-scale deep learning framework on FPGA for

fingerprint-based indoor localization, IEEE Access 8 (2020) 65609–65617. doi:10.1109/ACCESS.2020.2985162.

[23] P. Roy, C. Chowdhury, A survey on ubiquitous Wi-Fi-based indoor localization system for smartphone users from implementation perspectives, CCF Transactions on Pervasive Computing and Interaction 4 (2022) 298–318. doi:10.1007/s42486-022-00089-3.

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

[25] R. Mautz, Indoor positioning technologies, Thesis (2012). doi:10.3929/ethz-a-007313554.
URL http://hdl.handle.net/20.500.11850/54888

[26] F. Zafari, I. Papapanagiotou, K. Christidis, Microlocation for internet of things-equipped smart buildings, IEEE Internet of Things Journal 3 (1) (2016) 96–112. doi:10.1109/JIOT.2015.2442956.
URL https://ieeexplore.ieee.org/document/7120085/

[27] B. C, A. E. Khadimi, Survey on indoor localization system and recent dvances of Wi-Fi fingerprinting technique, in: 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), pp. 253–259. doi:10.1109/ICMCS.2016.7905633.

[28] W. Dargie, C. Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice, 2011. doi:10.1002/9780470666388.

[29] A. Nessa, B. Adhikari, F. Hussain, X. N. Fernando, A survey of machine learning for indoor positioning, IEEE Access 8 (2020) 214945–214965. doi:10.1109/ACCESS.2020.3039271.

[30] Z. Yang, Z. Zhou, Y. Liu, From RSSI to "CSI: Indoor localization via channel response, Acm Computing Surveys 46 (2) (2013) 1–32.

[31] L. Gui, M. Yang, P. Fang, S. Yang, RSS-based indoor localisation using MDCF, IET Wireless Sensor Systems 7 (4) (2017) 98–104. doi:10.1049/iet-wss.2016.0085.

URL https://ieeexplore.ieee.org/ielx7/5704589/7979714/07979716.pdf?tp=&arnumber=7979716&isnumber=7979714

[32] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.

URL https://doi.org/10.1038/nature14539https://www.nature.com/articles/nature14539.pdf

[33] W.-Y. Lin, C.-C. Huang, N.-T. Duc, H.-N. Manh, Wi-Fi indoor localization based on multi-task deep learning, in: 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018, pp. 1–5. doi:10.1109/ICDSP.2018.8631868.

[34] C.-H. Hsieh, J.-Y. Chen, B.-H. Nien, Deep learning-based indoor localization

using received signal strength and channel state information, IEEE Access 7 (2019) 33256–33267. doi:10.1109/ACCESS.2019.2903487.

[35] J. Zou, X. Guo, L. Li, S. Zhu, X. Feng, Deep regression model for received signal strength based Wi-Fi localization, in: 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018, pp. 1–4. doi:10.1109/ICDSP.2018.8631593.

[36] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, in: Advances in Neural Information Processing Systems, 2017, pp. 3856–3866.

[37] W. Njima, M. Chafii, R. M. Shubair, Gan based data augmentation for indoor localization using labeled and unlabeled data, in: 2021 International Balkan Conference on Communications and Networking (BalkanCom), 2021, pp. 36–39. doi:10.1109/BalkanCom53780.2021.9593240.

[38] J. Liu, N. Liu, Z. Pan, X. You, AutLoc: Deep autoencoder for indoor localization with RSS fingerprinting, in: 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), 2018, pp. 1–6. doi:10.1109/WCSP.2018.8555665.

[39] M. Yan, J. Wang, Z. Zhao, Online detection of Wi-Fi fingerprint alteration strength via deep learning, in: 2020 IEEE 45th Conference on Local Computer Networks (LCN), 2020, pp. 321–324. doi:10.1109/LCN48667.2020.9314835.

[40] L. Wang, S. Tiku, S. Pasricha, Chisel: Compression-aware high-accuracy em-

bedded indoor localization with deep learning, IEEE Embedded Systems Letters (2021) 1–1doi:10.1109/LES.2021.3094965.

[41] B. Chidlovskii, L. Antsfeld, Semi-supervised variational autoencoder for Wi-Fi indoor localization, in: 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2019.

[42] J. Xiao, Z. Zhou, Y. Yi, L. M. Ni, A survey on wireless indoor localization from the device perspective, ACM Computing Surveys (CSUR) 49 (2) (2016) 25.

[43] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, D. O. Wu, Indoor intelligent fingerprint-based localization: Principles, approaches and challenges, IEEE Communications Surveys & Tutorials 22 (4) (2020) 2634–2657.

[44] Z. He, Y. Li, L. Pei, R. Chen, N. El-Sheimy, Calibrating multi-channel RSS observations for localization using gaussian process, IEEE Wireless Communications Letters 8 (4) (2019) 1116–1119.

[45] M. Katwe, P. Ghare, P. K. Sharma, A. Kothari, NLOS error mitigation in hybrid RSS-ToA-based localization through semi-definite relaxation, IEEE Communications Letters 24 (12) (2020) 2761–2765.

[46] X. Tian, M. Wang, W. Li, B. Jiang, D. Xu, X. Wang, J. Xu, Improve accuracy of fingerprinting localization with temporal correlation of the RSS, IEEE Transactions on Mobile Computing 17 (1) (2018) 113–126. doi:10.1109/TMC.2017.2703892.

[47] J. Buckman, A. Roy, C. Raffel, I. Goodfellow, Thermometer encoding: One Hot way to resist adversarial examples, in: Proceedings of Sixth International Conference on Learning Representations (ICLR 2018), 2018.

[48] S. Sadowski, P. Spachos, RSSI-based indoor localization with the internet of things, IEEE Access 6 (2018) 30149–30161.

[49] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, A. Al-Fuqaha, Smart cities: A survey on data management, security, and enabling technologies, IEEE Communications Surveys & Tutorials 19 (4) (2017) 2456–2501.

[50] S. Alletto, R. Cucchiara, G. Del Fiore, L. Mainetti, V. Mighali, L. Patrono, G. Serra, An indoor location-aware system for an IoT-based smart museum, IEEE Internet of Things Journal 3 (2) (2015) 244–253.

[51] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent advances in indoor localization: A survey on theoretical approaches and applications, IEEE Communications Surveys & Tutorials 19 (2) (2016) 1327–1346.

[52] S. Xia, Y. Liu, G. Yuan, M. Zhu, Z. Wang, Indoor fingerprint positioning based on Wi-Fi: An overview, ISPRS International Journal of Geo-Information 6 (5) (2017) 135.

[53] M. Mahmud, M. S. Kaiser, A. Hussain, S. Vassanelli, Applications of deep

learning and reinforcement learning to biological data, IEEE Transactions on Neural Networks and Learning Systems 29 (6) (2018) 2063–2079.

[54] D. Li, B. Zhang, C. Li, A feature-scaling-based $k$-nearest neighbor algorithm for indoor positioning systems, IEEE Internet of Things Journal 3 (4) (2015) 590–597.

[55] J. Hong, T. Ohtsuki, Signal eigenvector-based device-free passive localization using array sensor, IEEE Transactions on Vehicular Technology 64 (4) (2015) 1354–1363.

[56] W. Zhang, K. Liu, W. Zhang, Y. Zhang, J. Gu, Deep neural networks for wireless localization in indoor and outdoor environments, Neurocomputing 194 (2016) 279–287.

[57] H. J. Bae, L. Choi, Large-scale indoor positioning using geomagnetic field with Deep Neural Networks, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), IEEE, 2019, pp. 1–6.

[58] G. E. Hinton, A. Krizhevsky, S. D. Wang, Transforming auto-encoders, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 44–51.

[59] C. Xiang, L. Zhang, Y. Tang, W. Zou, C. Xu, Ms-capsnet: A novel multi-scale capsule network, IEEE Signal Processing Letters 25 (12) (2018) 1850–1854.

[60] X. Wang, K. Tan, Q. Du, Y. Chen, P. Du, Caps-tripleGAN: GAN-assisted cap-

snet for hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing (2019).

[61] T. Iqbal, Y. Xu, Q. Kong, W. Wang, Capsule routing for sound event detection, in: 2018 26th European Signal Processing Conference (EUSIPCO), IEEE, 2018, pp. 2255–2259.

[62] L. Xiao, H. Zhang, W. Chen, Y. Wang, Y. Jin, Mcapsnet: Capsule network for text with multi-task learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4565–4574.

[63] C.-M. Own, J. Hou, W. Tao, Signal fuse learning method with dual bands Wi-Fi signal measurements in indoor positioning, IEEE Access 7 (2019) 131805–131817.

[64] M. Mohammadi, A. Al-Fuqaha, M. Guizani, J. Oh, Semi-supervised deep reinforcement learning in support of IoT and smart city services, IEEE Internet of Things Journal 5 (2) (2018) 624–635.

[65] A. Mittal, S. Tiku, S. Pasricha, Adapting convolutional neural networks for indoor localization with smart mobile devices, in: Proceedings of the 2018 on Great Lakes Symposium on VLSI, ACM, 2018, pp. 117–122.

[66] W. Xu, X. Fan, T. Wu, Y. Xi, P. Yang, C. Tian, Interest users cumulatively in your ads: A near optimal study for Wi-Fi advertisement scheduling, in: GI-2021-2021 IEEE Global Internet (GI) Symposium, IEEE, 2021, pp. 1–6.

[67] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. R. Sethi, D. Vasisht,

D. Bharadia, Deep learning based wireless localization for indoor navigation, in: Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, 2020, pp. 1–14.

[68] A. Belmonte-Hernández, G. Hernández-Peñaloza, F. Alvarez, G. Conti, Adaptive fingerprinting in multi-sensor fusion for accurate indoor tracking, IEEE Sensors Journal 17 (15) (2017) 4983–4998.

[69] J. Machaj, P. Brida, N. Majer, Challenges introduced by heterogeneous devices for Wi-Fi-based indoor localization, Concurrency and Computation: Practice and Experience (2019).

[70] Y. Li, S. Williams, B. Moran, A. Kealy, A probabilistic indoor localization system for heterogeneous devices, IEEE Sensors Journal 19 (16) (2019) 6822–6832.

[71] X. Tian, R. Shen, D. Liu, Y. Wen, X. Wang, Performance analysis of RSS fingerprinting based indoor localization, IEEE Transactions on Mobile Computing 16 (10) (2016) 2847–2861.

[72] L. Gong, C. Xiang, X. Fan, T. Wu, W. Yang, Device-free near-field human sensing using Wi-Fi signals (2020).
URL https://doi.org/10.1007/s00779-020-01385-4.

[73] H. Zou, B. Huang, X. Lu, H. Jiang, L. Xie, Standardizing location fingerprints across heterogeneous mobile devices for indoor localization, in: 2016 IEEE Wireless Communications and Networking Conference, 2016.

[74] X. Fan, X. He, C. Xiang, D. Puthal, L. Gong, P. Nanda, G. Fang, Towards system implementation and data analysis for crowdsensing based outdoor RSS maps, IEEE Access 6 (2018) 47535–47545.

[75] Z. He, Y. Li, L. Pei, R. Chen, N. El-Sheimy, Calibrating multi-channel RSS observations for localization using gaussian process, Wireless Communications Letters, IEEE 8 (4) (2019) 1116–1119.

[76] F. Furfari, A. Crivello, P. Baronti, P. Barsocchi, e. Michele Girolami, Discovering location based services: A unified approach for heterogeneous indoor localization systems, Internet of Things 13 (2021) 100334.

[77] H. Li, J. K. Ng, V. C. Cheng, W. K. Cheung, Fast indoor localization for exhibition venues with calibrating heterogeneous mobile devices, Internet of Things 3-4 (2018) 175–186.

[78] X. Fan, P. Yang, C. Xiang, L. Shi, imap: A crowdsensing based system for outdoor radio signal strength map, in: 2016 IEEE Trustcom/BigDataSE/ISPA, 2016.

[79] G. E. Hinton, S. Sabour, N. Frosst, Matrix capsules with EM routing, in: International Conference on Learning Representations, 2018.

[80] A. K. M. M. Hossain, Y. Jin, W. Soh, H. N. Van, SSD: A robust RF location fingerprint addressing mobile devices' heterogeneity, IEEE Transactions on Mobile Computing 12 (1) (2013) 65–77. doi:10.1109/TMC.2011.243.

[81] W. Li, Z. Chen, X. Gao, W. Liu, J. Wang, Multimodel framework for in-

door localization under mobile edge computing environment, IEEE Internet of Things Journal 6 (3) (2018) 4844–4853.

[82] G. Premsankar, M. D. Francesco, T. Taleb, Edge computing for the internet of things: A case study, IEEE Internet of Things Journal 5 (2) (2018) 1275–1284.

[83] B. Lashkari, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, Crowdsourcing and sensing for indoor localization in IoT: A review, IEEE Sensors Journal 19 (7) (2018) 2408–2434.

[84] L. Gong, Y. Zhao, C. Xiang, Z. Li, C. Qian, P. Yang, Robust light-weight magnetic-based door event detection with smartphones, IEEE Transactions on Mobile Computing 18 (11) (2018) 2631–2646.

[85] G. Félix, M. Siller, E. N. Alvarez, A fingerprinting indoor localization algorithm based deep learning, in: 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2016, pp. 1006–1011.

[86] K. S. Kim, S. Lee, K. Huang, A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting, Big Data Analytics 3 (1) (2018) 4. `doi:10.1186/s41044-018-0031-2`. URL `https://doi.org/10.1186/s41044-018-0031-2https://bdataanalytics.biomedcentral.com/track/pdf/10.1186/s41044-018-0031-2`

[87] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, J. Huerta, UJIIndoorLoc: A new multi-building

and multi-floor dataset for WLAN fingerprint-based indoor localization problems, in: Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on, IEEE, 2014, pp. 261–270.

[88] Y. Xiao, T. Ai, M. Yang, X. Zhang, A multi-scale representation of point-of-interest features in indoor map visualization, ISPRS International Journal of Geo-Information 9 (4) (2020).

[89] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, M. Youssef, Wideep: Wi-Fi-based accurate and robust indoor localization system using deep learning, in: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2019.

[90] D. Li, J. Xu, Z. Yang, Y. Lu, Q. Zhang, X. Zhang, Train once, locate anytime for anyone: Adversarial learning based wireless localization, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–10. doi:10.1109/INFOCOM42981.2021.9488693.

[91] L. Yuan, Y. Hu, Y. Li, R. Zhang, Y. Zhang, T. Hedgpeth, Secure RSS-fingerprint-based indoor positioning: Attacks and countermeasures, in: 2018 IEEE Conference on Communications and Network Security (CNS), 2018, pp. 1–9.

[92] C. Wang, J. Luo, X. Liu, X. He, Secure and reliable indoor localization based on multi-task collaborative learning for large-scale buildings, IEEE Internet of Things Journal (2021).

[93] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, Z. Han, Applications of economic and pricing models for wireless network security: A survey, IEEE Communications Surveys Tutorials 19 (4) (2017) 2735–2767.

[94] Y. Zeng, J. Cao, J. Hong, S. Zhang, L. Xie, Secure localization and location verification in wireless sensor networks: A survey, The Journal of Supercomputing 64 (3) (2013) 685–701.

[95] K. Järvinen, H. Leppäkoski, E.-S. Lohan, P. Richter, T. Schneider, O. Tkachenko, Z. Yang, PILOT: Practical privacy-preserving indoor localization using outsourcing, in: 2019 IEEE European Symposium on Security and Privacy, 2019, pp. 448–463.

[96] S. He, S.-H. G. Chan, Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons, IEEE Communications Surveys Tutorials 18 (1) (2016) 466–490.

[97] H. Li, J. K. Ng, V. C. Cheng, W. K. Cheung, Fast indoor localization for exhibition venues with calibrating heterogeneous mobile devices, Internet of Things 3-4 (2018) 175 – 186.

[98] M. Sugasaki, M. Shimosaka, Robust indoor localization across smartphone models with ellipsoid features from multiple RSSIs, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1 (3) (sep 2017).

[99] D. Li, J. Xu, Z. Yang, C. Wu, J. Li, N. D. Lane, Wireless localization with

spatial-temporal robust fingerprints, ACM Trans. Sen. Netw. 18 (1) (Oct. 2021).

[100] X. Fan, C. Xiang, L. Gong, X. He, Y. Qu, S. Amirgholipour, Y. Xi, P. Nanda, X. He, Deep learning for intelligent traffic sensing and prediction: Recent advances and future challenges, CCF Transactions on Pervasive Computing and Interaction (2020) 1–21.

[101] R. Elbakly, M. Youssef, The storyteller: Scalable building- and AP-independent deep learning-based floor prediction, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (1) (2020).

[102] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, C. J. Spanos, Adversarial learning-enabled automatic Wi-Fi indoor radio map construction and adaptation with mobile robot, IEEE Internet of Things Journal 7 (8) (2020) 6946–6954.

[103] X. Huang, G. Mei, J. Zhang, Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[104] Z. Shi, J. A. Zhang, R. Y. Xu, Q. Cheng, Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning, IEEE Transactions on Mobile Computing 21 (2) (2022) 540–554. doi:10.1109/TMC.2020.3012433.

[105] Y. Li, Y. Hu, R. Zhang, Y. Zhang, T. Hedgpeth, Secure indoor positioning against signal strength attacks via optimized multi-voting, in: 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS), pp. 1–10.

[106] M. Conti, N. Dragoni, V. Lesyk, A survey of man in the middle attacks, IEEE Communications Surveys Tutorials 18 (3) (2016) 2027–2051.

[107] N. Yang, X. Fan, D. Puthal, X. He, P. Nanda, S. Guo, A novel collaborative task offloading scheme for secure and sustainable mobile cloudlet networks, IEEE Access 6 (2018) 44175–44189.

[108] J. Li, M. Liu, Z. Xue, X. Fan, X. He, Rtvd: A real-time volumetric detection scheme for DDoS in the internet of things, IEEE Access 8 (2020) 36191–36201.

[109] J. Lv, D. Man, W. Yang, L. Gong, X. Du, M. Yu, Robust device-free intrusion detection using physical layer information of Wi-Fi signals, Applied Sciences 9 (1) (2019).

[110] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. P. Jayaraman, A. Y. Zomaya, Secure authentication and load balancing of distributed edge datacenters, Journal of Parallel and Distributed Computing 124 (2019) 60–69.

[111] Z. Meng, X. Zhan, J. Li, Z. Pan, An enhancement denoising autoencoder for rolling bearing fault diagnosis, Measurement 130 (2018) 448–454.

[112] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and com-

posing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ACM, pp. 1096–1103.

[113] X. Fan, C. Xiang, L. Gong, X. He, C. Chen, X. Huang, Urbanedge: Deep learning empowered edge computing for urban IoT time series prediction, in: Proceedings of the ACM Turing Celebration Conference-China, 2019, pp. 1–6.

[114] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size (2016).

[115] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017).

[116] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018, pp. 6848–6856.

[117] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[118] R. J. Wang, X. Li, C. X. Ling, Pelee: A real-time object detection system on mobile devices 31 (2018).

[119] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, Shufflenet v2: Practical guidelines for efficient cnn architecture design, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, Munich, Germany, 2018, pp. 116–131.

[120] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q. V. Le, Mnasnet: Platform-aware neural architecture search for mobile, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Long Beach, CA, USA, 2019, pp. 2820–2828.

[121] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, Seoul, Korea (South), 2019, pp. 1314–1324.

[122] S. Mehta, M. Rastegari, Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer (2021).

[123] Y. Fan, G. Zhao, X. Lei, W. Liang, K.-C. Li, K.-K. R. Choo, C. Zhu, SBBS: A secure blockchain-based scheme for IoT data credibility in fog environment, IEEE Internet of Things Journal 8 (11) (2021) 9268–9277.

[124] J. Li, M. Liu, Z. Xue, X. Fan, X. He, RTVD: A real-time volumetric detection scheme for DDoS in the Internet of Things, IEEE Access 8 (2020) 36191–36201.